



**SignalVu**  
**Vector Analysis Software**  
**Programmer Manual**

Register now!  
Click the following link to protect your product.  
[tek.com/register](https://tek.com/register)



077-0223-10 July 2024

Copyright © 2024, Tektronix. 2024 All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved. All other trade names referenced are the service marks, trademarks, or registered trademarks of their respective companies.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

Tektronix, Inc.  
14150 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
US

For product information, sales, service, and technical support visit [tek.com](https://www.tek.com) to find contacts in your area. For warranty information visit [tek.com/warranty](https://www.tek.com/warranty).

# Contents

Preface.....	5
Related documentation.....	5
Getting Started.....	6
Overview of the Manual.....	6
Setting up remote communication.....	7
Measurement support.....	7
Remote command support.....	10
Syntax and Commands.....	13
Command syntax.....	13
Backus-Naur Form Definition.....	13
SCPI Commands and Queries.....	13
IEEE 488.2 Common Commands.....	17
Constructed Mnemonics.....	18
Command groups.....	18
Functional groups.....	19
Programming hints.....	19
IEEE common commands.....	20
Obsolete commands.....	20
Abort commands.....	21
Calculate commands.....	21
Marker mnemonics.....	41
Calibration commands.....	41
Display commands.....	42
Fetch commands.....	64
Initiate commands.....	86
Input commands.....	86
Mass memory commands.....	87
Specifying the File.....	90
Output commands.....	90
Read commands.....	92
Sense commands.....	113
Status commands.....	143
Source commands.....	146
System commands.....	147
Trace commands.....	148
Trace mnemonics.....	156
Trigger commands.....	156
Unit commands.....	159
Retrieving response message.....	159
Command descriptions.....	161
Calculate commands.....	162
Calibration commands.....	518
Display commands.....	527
Fetch commands.....	974

Input commands.....	1444
Memory commands.....	1452
Output commands.....	1516
Read commands.....	1527
Sense commands.....	1950
Status commands.....	2658
System commands.....	2713
Trace Commands.....	2750
Trigger commands.....	2894
Status and Events.....	2935
Status and event reporting system.....	2935
Status Byte.....	2938
Standard event status block.....	2939
Operation status block.....	2940
Questionable Status Block.....	2941
Queues.....	2943
Status and event processing sequence.....	2943
Synchronizing execution.....	2944
Error messages and codes.....	2946
Command errors.....	2946
Execution errors.....	2947
Device specific errors.....	2949
Query Errors.....	2949
Device errors.....	2949
Status Conditions.....	2952
Global status.....	2952
Measurement status.....	2952
Source status.....	2953
Appendices.....	2955
Appendix A: Character charts.....	2955
Appendix B: SCPI conformance information.....	2955
Appendix C: Creating a DPX Frame Access Plugin.....	2957
Glossary.....	2960
Glossary.....	2960

---

# Preface

This document provides supplementary information about the remote commands for the SignalVu Vector Signal Analysis Software. The SignalVu software enables you to use the MSO/DPO5000, DPO7000, and DPO/DSA/MSO70000 Series Oscilloscopes to analyze RF signal behavior with the same software used on the RSA6000 Series Real-Time Analyzers. The remote command details are described in the following manuals that can be downloaded from [www.tektronix.com/manuals](http://www.tektronix.com/manuals).

- RSA6100B Series and RSA5100A Series Real-Time Spectrum Analyzers Programmer Manual
- MSO/DPO5000, DPO7000/C, DPO70000/B/C, DSA70000/B/C, and MSO70000/C Series Digital Oscilloscopes Programmer Manual

The SignalVu software supports the use of the RSA6100B Series commands with some limitations. For example, commands that control output functions are not supported. The details are given in the *Measurement Display Support* and *Remote Command Support* sections of this manual. Some commands are added for the SignalVu software and are described in the *Additional Commands* section. The error messages and codes are listed in the *Error Messages and Codes* section.

This manual is composed of the following sections:

- *Getting Started* outlines how to use the GPIB interface.
- *Syntax and Commands* defines the syntax used in command descriptions, presents a list of all command subsystems, and presents detailed descriptions of all programming commands.
- *Status and Events* describes how the status and Events Reporting system operates and presents a list of all system errors.
- *Appendices* provides additional information including character charts, GPIB interface specification, and factory initialization settings.

## Related documentation

- *SignalVu Vector Signal Analysis Software Help*

The help contains detailed information about how to operate the application and is found in the SignalVu application. A PDF of the Help is also available at [www.tek.com/manuals](http://www.tek.com/manuals).

- *TekVISA Programmer Manual*

(Tektronix part number 077-0140-XX)

This manual is available as a printable PDF file on the Tektronix Web site ([www.tek.com/manuals](http://www.tek.com/manuals)). The manual describes TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

# Getting Started

You can write computer programs that remotely take measurements and read those measurements for further analysis or storage. To help you get started with programming, this section includes the following subsections.

- Overview of the Manual  
Summarizes each major section of this manual.
- Measurement Support  
Describes which RSA measurement displays are available with the software.
- Setting Up Remote Communication  
Describes how to set up an oscilloscope to a controller and set a unique device address.
- Remote Command Support  
Describes which groups of RSA commands are supported or not supported by the software.

## Overview of the Manual

The information contained in each major section of this manual is described below.

### Syntax and Commands

*Syntax and Commands*, describes the structure and content of the messages your program sends to the analyzer. The following figure shows command parts as described in the *Command Syntax* subsection.

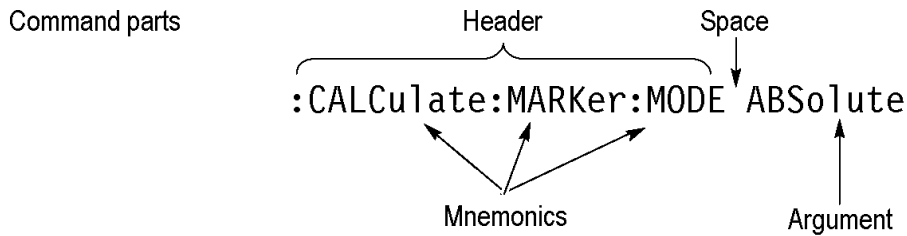


Figure 1: Command parts

Section 2 also describes the effect of each command and provides examples of how you might use it. The *Command Groups* subsection provides lists by functional areas. The commands are listed alphabetically in the *Command Descriptions* section.

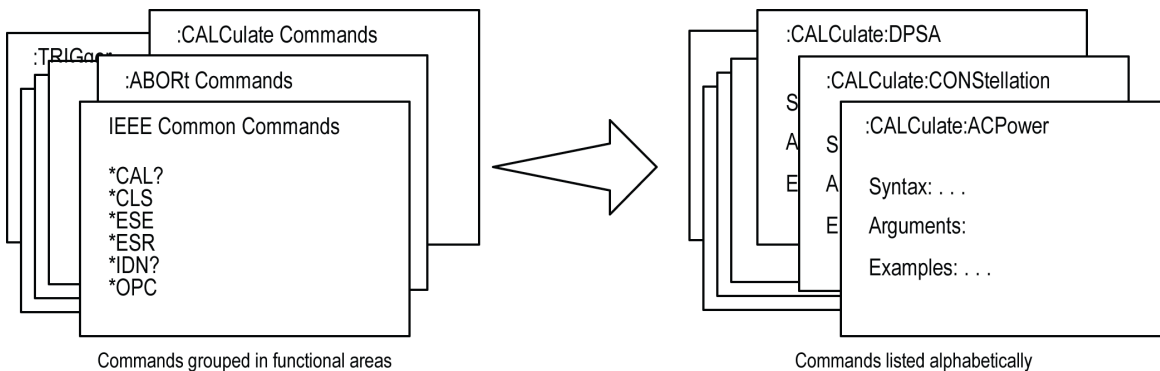


Figure 2: Functional groupings and an alphabetical list of commands

## Status and Events

The program may request information from the instrument. The instrument provides information in the form of status and error messages. The following figure illustrates the basic operation of this system. Section 3, Status and Events, describes how to get status or event information from the program and details the event and error messages.

Your program requests  
status and event reports. →

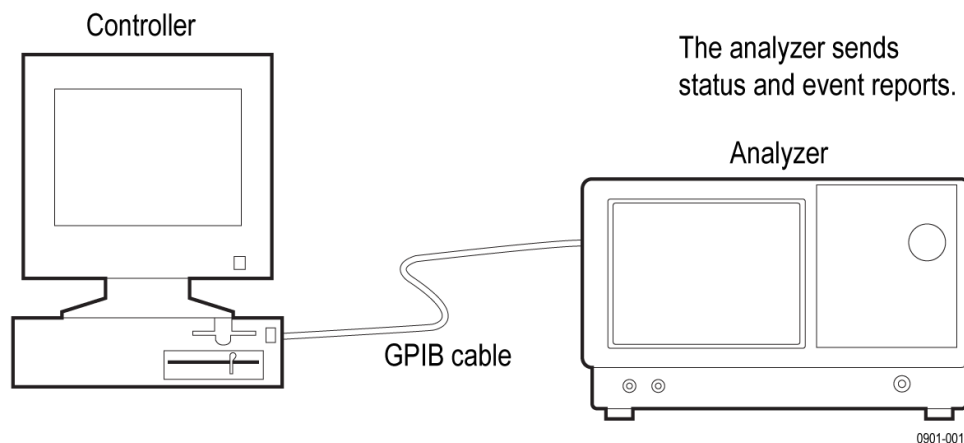


Figure 3: Event-driven program

## Setting up remote communication

Connect your oscilloscope to a controller and set a unique device address. The default settings for the GPIB configuration are:

- GPIB Address 1
- GPIB Mode GPIB Talk/Listen

To change either of the GPIB settings, do the following:

1. Select **GPIB Configuration** from the Utilities menu.
2. Click the **Configuration Talk/Listen** button.
3. Change the **GPIB Address** to a unique address.
4. Click the Close button.

The instrument is now set up for bidirectional communication with your controller. For details, refer to the Programmer manual for your oscilloscope.

## Measurement support

The RSA6100B Series analyzers support various measurements, which are categorized into measurement view groups. The following table lists all measurement displays available on the RSA6100B Series analyzers. The columns at right show whether or not a measurement is supported by the SignalVu software. Required options, if applicable, are noted in the left column.



**Note:** If you send a command for the measurement view that is not active on screen, an execution error will occur.



**Note:** Some measurement displays require installed options. See the Tektronix Web site product page for a list of available options.

**Table 1: Measurement support**

Measurement group	Measurement
General signal viewing	Amplitude vs Time
	Frequency vs Time
	Phase vs Time
	RF I&Q vs Time
	Spectrogram
	Spectrum
	Time overview
Analog Modulation	Amplitude modulation
	Frequency modulation
	Phase modulation
Phase Noise and Jitter	Phase Noise Display
Settling Time (requires Option SV12)	Freq Settling
	Phase Settling
General purpose digital modulation (requires Option SV21)	Constellation
	Demod I&Q versus Time
	EVM vs Time
	Eye Diagram
	Freq Dev vs Time
	Mag Error vs Time
	Phase Error vs Time
	Signal quality
	Symbol table
	Trellis
User defined filters	
RF	Channel power and ACPR
	Freq Settling (requires Option SVT).
	MCPR
	Occupied Bandwidth
	Phase Settling (requires Option SVT)
	Signal Strength (requires Option MAP)
	SEM
	Spurious

Table continued...



Measurement group	Measurement
OFDM(requires Option SVO)	OFDM Channel Response
	OFDM Constellation
	OFDM EVM
	OFDM Mag Error
	OFDM Phase Error
	OFDM Power
	OFDM Summary
	OFDM Symbol Table
	OFDM Flatness
WLAN (requires Option SV23,SV24, or SV25)	WLAN Channel Response
	WLAN Constellation
	WLAN EVM
	WLAN Flatness
	WLAN Mag Error
	WLAN Phase Error
	WLAN Power vs Time
	WLAN Summary
	WLAN Symbol Table
	SEM
P25 Analysis (requires Option SV26)	P25 Eye Diagram
	P25 Freq Dev vs Time
	P25 Power vs Time
	P25 Constellation
	P25 Symbol Table
	P25 Summary
	MCPR
	Time Overveiw
Bluetooth Analysis (requires Option SV27)	BT 20dB BW
	BT CF Offset and Drift
	BT Constellation
	BT Eye Diagram
	BT Freq Dev vs Time
	BT Inband Emission
	BT Summary
	BT Symbol Table
	Time Overview

Table continued...

Measurement group	Measurement
LTE Analysis (requires Option SV28)	LTE ACLR
	LTE Channel Spectrum
	LTE Constellation
	LTE Power vs Time
	SEM
Pulsed RF (requires Option SVP)	Pulse statistics
	Pulse table
	Pulse trace
802.11ad Analysis (requires Option SV30)	802.11ad Summary
	802.11ad Symbol Table
	802.11ad Constellation
	802.11ad EVM vs Time
5G NR Analysis (requires Option SVE and 5G NR)	NR Constellation
	NR Channel Power
	NR Adjacent Channel Power
	NR Spectral Emission Mask
	NR Summary
Audio Analysis (requires Option SVA)	Audio Spectrum
	Audio Summary

## Remote command support

The SignalVu software supports the use of the RSA6100B Series analyzer remote commands with some limitations. The following table summarizes the RSA6100B Series commands supported and not supported by the SignalVu software for each command group. For details on the RSA6100B Series remote commands, refer to the RSA6100B Programmer Manual. It is available for download at [www.tektronix.com/downloads](http://www.tektronix.com/downloads).

**Table 2: Command support summary**

Command group	Function	Commands supported	Commands not supported
IEEE common	Conforms to the IEEE Std 488.2.	Use the oscilloscope IEEE	common commands. <sup>1</sup>
CALCulate	Controls the markers and the search operations.	All	CALCulate:DPSSA:

Table continued...

Command group	Function	Commands supported	Commands not supported
CALibration	Calibrates the instrument.	Use the oscilloscope	CALibrate commands. <sup>1</sup>
DISPlay	Controls the presentation of text, graph, and trace.	All	DISPlay:DPSA: DISPlay:CCDF:
FETCh	Retrieves the measurements from the last acquired data.	All except in the right box.	FETCh:RFIN:IQ:SCALE? FETCh:DPSA:
INITiate	Controls the initiation of data acquisition.	All	
INPut	Controls the characteristics of the signal input.	All except in the right box.	INPut[:RF]:ATTenuation INPut[:RF]:ATTenuation:AUTO INPut[:RF]:ATTenuation:MONitor:STATe INPut[:RF]:GAIN:STATe
MMEMory	Provides mass storage capabilities for the instrument.  Setup and TIQ files (:MMEMory:xxx:IQ and :MMEMory:xx:STATe) contain complete oscilloscope setup information. When one of these files is loaded, the oscilloscope is setup according to the saved oscilloscope settings in the file.	All except in the right box. (See <i>Note</i> following this table.)  You can also use the FILESystem commands of the oscilloscope. <sup>[1]</sup>	MMEMory:CALibration:LOAD:CORRection :EXtErnal:EDIT<x> MMEMory:CALibration:STORE:CORRection :EXtErnal:EDIT<x> MMEMory:DPSA:
OUTPut	Controls the characteristics of the signal output.	-	All
READ	Obtains the measurement results by acquiring fresh data.	All except in the right box.	READ:DPSA

Table continued...

Command group	Function	Commands supported	Commands not supported
SENSe	Sets up detailed conditions for each measurement.	All except in the right box. (See <i>Note</i> following this table.) You can also use the additional commands.	[SENSe]:ACQuisition:FFRame:ACTual? [SENSe]:ACQuisition:FFRame:Limit [SENSe]:ACQuisition:FFRame:STATe [SENSe]:ANALySis:ADVanced:DITher [SENSe]:ROSCillator:SOURce [SENSe]:DPSA:
STATus	Controls the status and event registers.	-	All
SYSTem	Sets or queries system parameters for operation.	All except in the right box.	SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS SYSTem:DATE SYSTem:TIME SYSTem:VERSion SYSTem:HW:VERSion
TRACe	Controls trace activation and math operations.	All except in the right box.	TRACe:DPSA
TRIGger	Controls triggering.	Use the oscilloscope	TRIGger commands. <sup>1</sup>
UNIT	Specifies fundamental units for measurement.	-	- UNIT:POWer (Use the [SENSe]:POWer:UNITs command.)

[1] Refer to the *MSO/DPO5000, DPO7000/C, DPO70000/B/C, DSA70000/B/C, and MSO70000/C Series Digital Oscilloscopes Programmer Manual..*

## Quick Tips

- The :MMEMory:LOAD:IQ command accepts the file extensions other than .tiq. In the RSA6100B Series spectrum analyzers, this command ignores the file extension passed in (if any), and assumes that it is .tiq. In the SignalVu software, the file extension can be either .tiq or .wfm, and so is not ignored. If no file extension is included, it is assumed to be .tiq.
- To avoid conflicts with the oscilloscope commands, use the following syntax.
  - Use the command header [SENSe]:SIGNAlvu:ACQuisition for the [SENSe]:ACQuisition commands. For example, replace the [SENSe]:ACQuisition:BANDwidth command with [SENSe]:SIGNAlvu:ACQuisition:BANDwidth.
  - Replace the [SENSe]:MEASurement:FREQuency command with [SENSe]:FREQuency.

# Syntax and Commands

## Command syntax

This section contains information on the Standard Commands for Programmable Instruments (SCPI) and IEEE 488.2 Common Commands you can use to program your RSA Series analyzer. The information is organized in the following subsections

- Backus-Naur Form Definition
- SCPI Commands and Queries
- IEEE 488.2 Common Commands
- Constructed Mnemonics

## Backus-Naur Form Definition

This manual may describe commands and queries using the Backus-Naur Form (BNF) notation. The following table defines the standard BNF symbols.

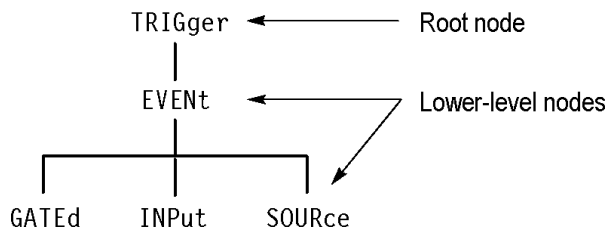
**Table 3: BNF symbols and meanings**

Symbol	Meaning
< >	Defined element
:=	Is defined as
	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
...	Previous element(s) may be repeated
( )	Comment

## SCPI Commands and Queries

SCPI is a standard created by a consortium that provides guidelines for remote programming of instruments. These guidelines provide a consistent programming environment for instrument control and data transfer. This environment uses defined programming messages, instrument responses, and data format across all SCPI instruments, regardless of manufacturer. The analyzer uses a command language based on the SCPI standard.

The SCPI language is based on a hierarchical or tree structure as shown in the following figure that represents a subsystem. The top level of the tree is the root node; it is followed by one or more lower-level nodes.



*Figure 4: Example of SCPI subsystem hierarchy tree*

You can create commands and queries from these subsystem hierarchy trees. Commands specify actions for the instrument to perform. Queries return measurement data and information about parameter settings.

### Creating Commands

SCPI commands are created by stringing together the nodes of a subsystem hierarchy and separating each node by a colon.

In the figure above, TRIGger is the root node and EVENT, GATed, INPut, and SOURce are lower-level nodes. To create a SCPI command, start with the root node TRIGger and move down the tree structure adding nodes until you reach the end of a branch. Most commands and some queries have parameters; you must include a value for these parameters. If you specify a parameter value that is out of range, the parameter will be set to a default value. The command descriptions list the valid values for all parameters.

For example, TRIGgerEVENT:SOURce EXTRear is a valid SCPI command created from the hierarchy tree. (See Figure 2.)

### Creating Queries

To create a query, start at the root node of a tree structure, move down to the end of a branch, and add a question mark. TRIGgerEVENT:SOURce? is an example of a valid SCPI query using the hierarchy tree in the figure. (See Figure 2.)

### Query Responses

The query causes the analyzer to return information about its status or settings. When a query is sent to the analyzer, only the values are returned. When the returned value is a mnemonic, it is noted in abbreviated format, as shown in the following table.

**Table 4: Query response examples**

Query	Response
CALCulate:SPECTrum:MARKer:X	7.50E+9
SENSe:SPECTrum:FFT:WINDow	BH4B

A few queries also initiate an operation action before returning information. For example, the \*CAL? query runs a calibration.

### Parameter Types

Every parameter in the command and query descriptions is of a specified type. The parameters are enclosed in brackets, such as <value>. The parameter type is listed after the parameter and is enclosed in parentheses, for example, (Boolean). Some parameter types are defined specifically for the RSA Series command set and some are defined by ANSI/IEEE 488.2-1987 as shown in the following table.

**Table 5: Parameter types used in syntax descriptions**

Parameter type	Description	Example
arbitrary block <sup>1</sup>	A specified length of arbitrary data	#512234xxxx . . . where 5 indicates that the following 5 digits (12234) specify the length of the data in bytes;xxxxx ... indicates the data
Boolean	Boolean numbers or values	ON or 1; OFF or 0
binary	Binary numbers	#B0110
octal	Octal numbers	#Q57, #Q3
hexadecimal <sup>2</sup>	Hexadecimal numbers (0-9, A, B, C, D, E, F)	#HAA, #H1

Table continued...

<sup>1</sup> Defined in ANSI/IEEE 488.2 as "Definite Length Arbitrary Block Response Data."

<sup>2</sup> An ANSI/IEEE 488.2-1992-defined parameter type.

Parameter type	Description	Example
NR1 <sup>2</sup> numeric	Integers	0, 1, 15, -1
NR2 <sup>23</sup> numeric	Decimal numbers	1.2, 3.141516, -6.5
NR3 <sup>2</sup>	Floating point numbers	3.1415E-9, -16.1E5
NRf <sup>2</sup> numeric	Flexible decimal number that may be type NR1, NR2 or NR3	See NR1, NR2, and NR3 examples
string <sup>4</sup>	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

## Special Characters

The Line Feed (LF) character (ASCII 10), and all characters in the range of ASCII 127-255 are defined as special characters. These characters are used in arbitrary block arguments only; using these characters in other parts of any command yields unpredictable results.

## Abbreviating Commands, Queries, and Parameters

You can abbreviate most SCPI commands, queries, and parameters to an accepted short form. This manual shows these short forms as a combination of upper and lower case letters. The upper case letters indicate the accepted short form of a command. As shown in the following figure, you can create a short form by using only the upper case letters. The accepted short form and the long form are equivalent and request the same action of the instrument.

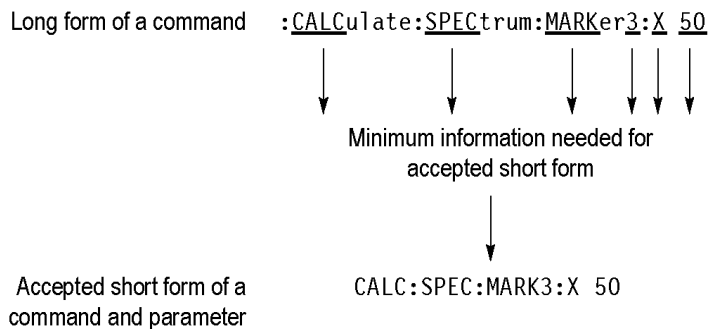


Figure 5: Example of abbreviating a command



**Note:** The numeric suffix of a command or query may be included in either the long form or short form; the analyzer will default to "1" if no suffix is used. In the above figure, the "3" of "MARKer3" indicates that the command is directed to Marker 3.

## Chaining Commands and Queries

You can chain several commands or queries together into a single message. To create a chained message, first create a command or query, add a semicolon(;), and then add more commands or queries and semicolons until the message is complete. If the command following a semicolon is a root node, precede it with a colon (:). The following figure illustrates a chained message consisting of several commands and queries. The single chained message should end in a command or query, not a semicolon. Responses to any queries in your message are separated by semicolons.

<sup>3</sup> Some commands and queries will accept an octal or hexadecimal value even though the parameter type is defined as NR1.

<sup>4</sup> Defined in ANSI/IEEE 488.2 as "String Response Data."

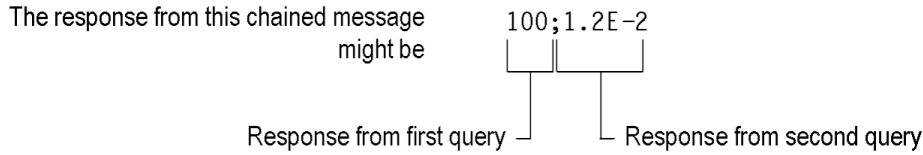
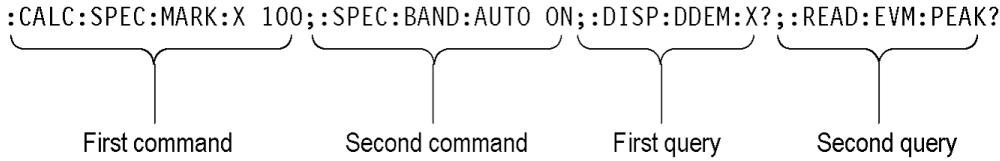


Figure 6: Example of chaining commands and queries

If a command or query has the same root and lower-level nodes as the previous command or query, you can omit these nodes. In the following figure, the second command has the same root node (TRIGgerEVENT) as the first command, so these nodes can be omitted.

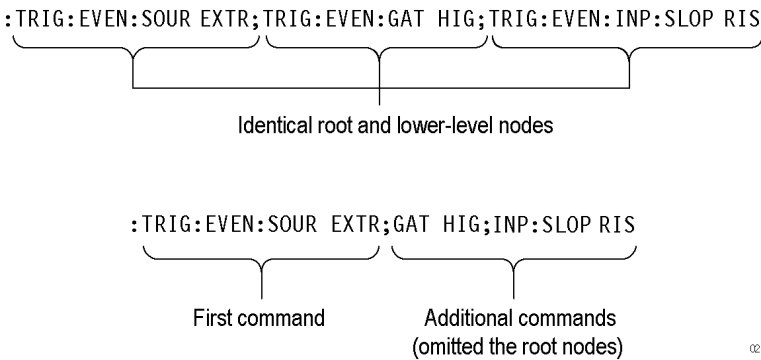


Figure 7: Example of omitting root and lower-level nodes in a chained message

### Unit and SI Prefix

If the decimal numeric argument refers to amplitude, frequency, or time, you can express it using SI units instead of using the scaled explicit point input value format <NR3>. (SI units are units that conform to the Systeme International d'Unites standard.) For example, you can use the input format 200 mV or 1.0 MHz instead of 200.0E-3 or 1.0E+6, respectively, to specify voltage or frequency.

The following table lists the available units.

**Table 6: Available units**

Symbol	Meaning
dB	decibel (relative amplitude)
dBm	decibel (absolute amplitude)
DEG	degree (phase)
Hz	hertz (frequency)
PCT	percent (%)
s	second (time)
V	volt

The available SI prefixes are shown in the following table.



**Table 7: Available SI prefixes**

SI prefix	Z	A	F	P	N	U	M	K	MA <sup>5</sup>	G	T	PE	EX
Corresponding power	10 <sup>-21</sup>	10 <sup>-18</sup>	10 <sup>-15</sup>	10 <sup>-12</sup>	10 <sup>-9</sup>	10 <sup>-6</sup>	10 <sup>-3</sup>	10 <sup>+3</sup>	10 <sup>+6</sup>	10 <sup>+9</sup>	10 <sup>+12</sup>	10 <sup>+15</sup>	10 <sup>+18</sup>

You can omit a unit in a command, but you must include the unit when using a SI prefix. For example, frequency of 15 MHz can be described as follows

15.0E6, 1.5E7Hz, 15000000, 15000000Hz, 15MHz, etc  
("15M" is not allowed.)

Note that you can use either lower or upper case units and prefixes. The following examples have the same result, respectively.

170mHz, 170MHz, 170MHz, etc  
250mv, 250mV, 250MV, etc

## General Rules

Here are three general rules for using SCPI commands, queries, and parameters:

- You can use single ( ' ) or double ( " ") quotation marks for quoted strings, but you cannot use both types of quotation marks for the same string.

`correct "This string uses quotation marks correctly."`

`correct 'This string also uses quotation marks correctly.'`

`incorrect "This string does not use quotation marks correctly.'`

- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.

`SENSE:SPECTRUM:FFT:LENGTH 1024`

is the same as

`sense:spectrum:fft:length 1024`

and

`SENSE:spectrum:FFT:length 1024`



**Note:** *Literal strings (quoted) are case sensitive, for example, file names.*

- No embedded spaces are allowed between or within nodes.

`correct SENSE:SPECTRUM:FFT:LENGTH 1024`

`incorrect SENSE: SPECTRUM: FFT: LEN GTH 1024`

## IEEE 488.2 Common Commands

ANSI/IEEE Standard 488.2 defines the codes, formats, protocols, and usage of common commands and queries used on the interface between the controller and the instruments. The analyzer complies with this standard.

<sup>5</sup> When the unit is "Hz", "M" may be used instead of "MA" so that the frequency can be represented by "MHz"

## Command and Query Structure

The syntax for an IEEE 488.2 common command is an asterisk (\*) followed by a command and, optionally, a space and parameter value. The syntax for an IEEE 488.2 common query is an asterisk (\*) followed by a query and a question mark. All of the common commands and queries are listed in the last part of the `Syntax and Commands` section. The following are examples of common commands:

- \*ESE 16
- \*CLS

The following are examples of common queries

- \*ESR
- \*IDN

## Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a trace mnemonic can be either TRACe1, TRACe2, TRACe3, or TRACe4. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a TRACe1:SPECTrum:FUNCTion command, and there is also a TRACe2:SPECTrum:FUNCTion command. In the command descriptions, this list of choices is abbreviated as TRACe<x>. The value of <x> is the upper range of valid suffixes. If the numeric suffix is omitted, the analyzer uses the default value of "1".

**Table 8: Constructed mnemonics**

Symbol	Meaning
MARKer<x>	A marker specifier where <x> = 0, 1, 2, 3, or 4. Refer to Marker Mnemonics
RANGe<x>	A range specifier where <x> = 1 to 20. Refer to [SENSE]:SPURious Subgroup for details.
SPUR<x>	A spurious specifier where <x> = 1 to the number of spurious signals. Refer to FETCh :READ:SPURious Subgroup for details, respectively.
TRACe<x>	A trace specifier where <x> = 0, 1, 2, 3, 4, or 5. Refer to TRACe Commands for details.

## Command groups

This section lists the analyzer commands in two ways. It first presents them by functional groups. It then lists them alphabetically. The functional group list starts below. The alphabetical list provides more detail on each command. For the complete list of commands, see the Command Descriptions section.

The analyzers conform to the Standard Commands for Programmable Instruments (SCPI) 1999.0 and IEEE Std 488.2-1987, except where noted.

Items followed by question marks are queries; items without question marks are commands. Some items in this section have a question mark in parentheses () in the command header section; this indicates that the item can be both a command and a query.

For the conventions of notation in this manual, refer to Command Syntax and following pages.

## Functional groups

All commands are divided into groups as shown in the following table.

**Table 9: List of command group**

Command group	Function
IEEE common	Conforms to the IEEE Std 488.2.
ABORt	Resets the trigger system and stops measurements.
CALCulate	Controls the markers and the search operations.
DISPlay	Controls the display of measurement results and waveforms.
FETCh	Retrieves the measurements from the last acquired data.
INITiate	Controls data acquisition.
INPut	Controls the characteristics of the signal input.
MMEMory	Provides mass storage capabilities for the analyzer.
OUTPut	Controls the characteristics of the signal output.
READ	Obtains the measurement results with acquiring data.
SENSe	Sets up detailed conditions for each measurement.
STATus	Controls the status and event registers.
SYSTem	Sets or queries system parameters for operation.
TRACe	Controls trace activation and math operations.
UNIT	Specifies fundamental units for measurement.

## Programming hints

Here are some basic tips for using commands:

- *Selecting a measurement item*

Use Display commands to select or display the measurement view. [Example] `DISPlay:GENeral:MEASview:NEW SPECTrum` Displays the spectrum view on the screen.

- *Setting measurement parameters*

Use Sense commands to set conditions for the measurement session. [Example] `SENSe:SPECTrum:FREQuency:CENTer 1.5GHz` Sets the center frequency to 1.5 GHz in the spectrum view.

- *Processing waveforms arithmetically*

Use Trace commands for math operation on waveforms. [Example] `TRACe1:SPECTrum:FUNCTion AVERAge` Averages the spectrum waveform.

- *Measuring with the markers*

Use Calculate commands to measure some quantity using the markers. [Example] `CALCulate:SPECTrum:MARKer1:MAXimum` Positions the marker at the highest peak signal on the spectrum.

- *Scaling the waveform*

Use Display commands to change the waveform portion on screen. [Example] `DISPlay:IQVTime:Y:SCALE 1.5` Sets the vertical range to 1.5 V in the IQ versus Time graph.

The following sections list the commands by group.

## IEEE common commands

The IEEE 488.2 common commands have a "\*" prefix.

**Table 10: Status and error commands**

Command	Description
*CLS	Clears status.
*ESE	Sets or queries the bits in the ESER register.
*ESR?	Returns the contents of the SESR register.
*IDN?	Returns the instrument identification code.
*OPC	Synchronizes commands.
*OPT?	Returns a list of options installed in your analyzer.
*RST	Returns the instrument settings to the factory defaults.
*SRE	Sets or queries the bits in the SRER register.
*STB?	Returns the contents of the SBR using the MSS bit.
*WAI	Prevents the analyzer from executing further commands.

## Obsolete commands

The following table lists obsolete commands and their replacements.

**Table 11: Obsolete and replacement commands**

Obsolete commands	Replacement commands
[SENSe:]PULSe:ANALyze:MEASurement:TIME:AUTO	SENSe:PULSe:MEASview<y>:ANALyze:FDOMain:MEASurement:TIME:ALENght
[SENSe:]PULSe:ANALyze:MEASurement:TIME:START	SENSe:PULSe:MEASview<y>:ANALyze:FDOMain:MEASurement:TIME:METHod

Table continued...

Obsolete commands	Replacement commands
[SENSe:]PULSe:ANALyze:MEASurement:TIME:STOP	SENSe:PULSe:MEASview<y>:ANALyze:FDOMain:MEASurement:TIME:RLEngth
	SENSe:PULSe:MEASview<y>:ANALyze:FDOMain:MEASurement:TIME:RLEVel
	SENSe:PULSe:MEASview<y>:ANALyze:FDOMain:MEASurement:TIME:START
	SENSe:PULSe:MEASview<y>:ANALyze:IRESpone:MEASurement:TIME:ALEngth
	SENSe:PULSe:MEASview<y>:ANALyze:IRESpone:MEASurement:TIME:METHOD
	SENSe:PULSe:MEASview<y>:ANALyze:IRESpone:MEASurement:TIME:RLEngth
	SENSe:PULSe:MEASview<y>:ANALyze:IRESpone:MEASurement:TIME:RLEVel
	SENSe:PULSe:MEASview<y>:ANALyze:IRESpone:MEASurement:TIME:START
	SENSe:PULSe:MEASview<y>:ANALyze:OVERshoot:MEASurement:TIME:ALEngth
	SENSe:PULSe:MEASview<y>:ANALyze:OVERshoot:MEASurement:TIME:METHOD
	SENSe:PULSe:MEASview<y>:ANALyze:OVERshoot:MEASurement:TIME:RLEngth
	SENSe:PULSe:MEASview<y>:ANALyze:OVERshoot:MEASurement:TIME:RLEVel
	SENSe:PULSe:MEASview<y>:ANALyze:OVERshoot:MEASurement:TIME:START
	[SENSe:]PULSe:FREFerence:CHIRpbw(?)



**Note:** <y> is the Measurement Index. Range: 1–8 (based on number of sources of connected instrument), where MEASview<y> keyword is optional and by default y is 1

## Abort commands

Use the Abort commands to reset the trigger system and to stop measurements.

**Table 12: Abort commands**

Command	Description
ABORt	Resets the trigger system and stops measurements.

## Calculate commands

Use the Calculate commands to control the markers and the search operations. See the Index for commands not listed in the following table.

**Table 13: Calculate commands**

Command	Description
CALCulate:MARKer:ADD	Adds a marker.
<b>CALCulate:TXGain subcommand group</b>	<b>Transmission Gain measurements</b>
CALCulate:TXGain:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the specified marker on the trace.
CALCulate:TXGain:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the specified marker on the trace.

Table continued...

Command	Description
CALCulate:TXGain:MARKer<x>:MAXimum	Moves the specified marker to the highest peak on the trace.
CALCulate:TXGain:MARKer<x>:PEAK:HIGHer	Moves the specified marker to the next peak higher in amplitude on the trace.
CALCulate:TXGain:MARKer<x>:PEAK:LEF	Moves the specified marker to the next peak to the left on the trace.
CALCulate:TXGain:MARKer<x>:PEAK:LOWer	Moves the specified marker to the next peak lower in amplitude on the trace.
CALCulate:TXGain:MARKer<x>:PEAK:RIGHT	Moves the specified marker to the next peak to the right on the trace.
CALCulate:TXGain:MARKer<x>:TRACe	Sets or queries the trace on which the specified marker is placed in the Transmission Gain display.
CALCulate:TXGain:MARKer<x>:X	Sets or queries the horizontal position for the specified marker on the trace.
CALCulate:TXGain:MARKer<x>:Y?	Queries the marker amplitude of the specified marker on the trace.
<b>CALCulate:BLUetooth subcommand group</b>	<b>Bluetooth measurements</b>
CALCulate:BLUetooth:CONStE:MARKer<x>:DELTA:X[:TIME]?	Queries the delta marker time for the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:FDEVIation?	Queries the frequency deviation for the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:MAGNitude?	Queries the magnitude readout of the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:MAXimum	Positions the selected marker at the symbol in the center of the analyzed time record.
CALCulate:BLUetooth:CONStE:MARKer<x>:PEAK:LEFT	Moves the selected marker in the time domain to the next symbol number to the left, relative to the previous marker position.
CALCulate:BLUetooth:CONStE:MARKer<x>:PEAK:RIGHT	Moves the selected marker in the time domain to the next symbol number to the right, relative to the previous marker position.
CALCulate:BLUetooth:CONStE:MARKer<x>:PHASe?	Queries the phase readout of the selected marker
CALCulate:BLUetooth:CONStE:MARKer<x>:SYMBol?	Queries the symbol readout value for the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:Trace	Sets or queries the Trace for the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:VALue?	Queries the value readout of the selected marker.
CALCulate:BLUetooth:CONStE:MARKer<x>:X	Sets or queries the time position of the selected marker.
CALCulate:BLUetooth:EDIagram:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:BLUetooth:EDIagram:MARKer<x>:DELTA:Y?	Returns the delta marker time for the selected marker.
CALCulate:BLUetooth:EDIagram:MARKer<x>:MAXimum	Returns the delta marker amplitude for the selected markers.
CALCulate:BLUetooth:EDIagram:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the highest peak.
CALCulate:BLUetooth:EDIagram:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left.
CALCulate:BLUetooth:EDIagram:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:BLUetooth:EDIagram:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right.
CALCulate:BLUetooth:EDIagram:MARKer<x>:TRACe	The command places the selected marker on the I or Q trace in the Bluetooth eye diagram display. The query returns the name of the trace where the current marker resides.
CALCulate:BLUetooth:EDIagram:MARKer<x>:X[:TIME]	Sets or queries the horizontal position (time) of the selected marker.
CALCulate:BLUetooth:EDIagram:MARKer<x>:Y?	Returns the vertical position of the selected marker.

Table continued...

Command	Description
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:DELTA:X[:TIME]?	Queries the delta marker time for the selected marker.
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker .
CALCulate:BLUEtooth:FDVTime:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace.
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude.
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace .
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:X[:TIME]	Sets or queries the marker time.
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:Y	Sets or queries the marker amplitude.
CALCulate:BOBW:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:BOBW:MARKer<x>:DELTA:Y?	Returns the delta marker frequency for the selected marker on the Bluetooth 20 dB bandwidth measurement.
CALCulate:BOBW:MARKer<x>:MAXimum	Positions the selected marker at the maximum value of the display.
CALCulate:BOBW:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude.
CALCulate:BOBW:MARKer<x>:PEAK:LEFT	Moves the selected marker in the time domain to the left, relative to the previous marker position on the trace.
CALCulate:BOBW:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:BOBW:MARKer<x>:PEAK:RIGHT	Moves the selected marker in the time domain to the right, relative to the previous marker position on the trace.
CALCulate:BOBW:MARKer<x>[:SET]:CENTer	Sets the center frequency to the value at the marker position.
CALCulate:BOBW:MARKer<x>:X	Sets or queries the time position of the selected marker.
CALCulate:BOBW:MARKer<x>:Y?	Returns the amplitude position of the selected marker.
<b>CALCulate:DPX:MARKer subcommand group</b>	<b>DPX power density readings</b>
CALCulate:DPX:MARKer<x>:POWER:DENSITY?	Returns the power density of the specified marker.
CALCulate:DPX:MARKer<x>:POWER:IDensity?	Returns the integrated power density reading between the MR (reference marker) and the selected marker.
CALCulate:DPX:MARKer<x>:POWER:INTEgrated?	Returns the integrated power reading between the MR (reference marker) and the selected marker.
CALCulate:MARKer:MODE	Sets or queries the marker mode.
<b>CALCulate:NOISE subcommand group</b>	<b>Noise and Gain measurements</b>
CALCulate:NOISE:FIGure:MARKer<x>:DELTA:X?	Queries the frequency difference between MR (reference marker) and the selected marker
CALCulate:NOISE:FIGure:MARKer<x>:DELTA:Y?	Queries the power difference value between MR (reference marker) and the selected marker.
CALCulate:NOISE:FIGure:MARKer<x>:MAXimum	Places the selected marker at the maximum value on the peak of the trace.
CALCulate:NOISE:FIGure:MARKer<x>:PEAK:HIGHer	Places the selected marker at the next highest peak on the trace.
CALCulate:NOISE:FIGure:MARKer<x>:PEAK:LEFT	Places the selected marker at the next peak on the trace to the left of the selected marker

Table continued...

Command	Description
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer	Places the selected marker at the next lowest peak on the trace.
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT	Places the selected marker at the next peak on the trace to the right of the selected marker.
CALCulate:NOISe:FIGure:MARKer<x>:TRACe	Sets or queries which trace the specified marker is attached to.
CALCulate:NOISe:FIGure:MARKer<x>:X?	Queries the frequency of the specified marker.
CALCulate:NOISe:FIGure:MARKer<x>:Y?	Queries the vertical position (dB) of the specified marker.
CALCulate:NOISe:GAIN:MARKer<x>:DELTA:X?	Query returns the frequency difference between MR (reference marker) and the selected marker.
CALCulate:NOISe:GAIN:MARKer<x>:DELTA:Y?	Query returns the power difference value between MR (reference marker) and the selected marker.
CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?	Places the selected marker at the maximum value on the peak of the trace.
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?	Places the selected marker at the next highest peak on the trace.
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT?	Places the selected marker at the next peak on the trace to the left of the selected marker.
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer?	Places the selected marker at the next lowest peak on the trace.
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?	Places the specified marker on a specified trace. The query form returns the trace on which the marker resides.
CALCulate:NOISe:GAIN:MARKer<x>:TRACe	Sets or queries which trace the specified marker is attached to.
CALCulate:NOISe:GAIN:MARKer<x>:X?	Queries the horizontal position of the specified marker.
CALCulate:NOISe:GAIN:MARKer<x>:Y?	Queries the vertical position of the specified marker.
CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:X?	Returns the frequency difference between MR (reference marker) and the selected marker.
CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:Y?	Returns the temperature difference between MR (reference marker) and the selected marker.
CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?	Places the selected marker at the maximum value on the peak of the trace.
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?	Places the selected marker at the highest peak on the trace.
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?	Places the selected marker at the next peak on the trace to the left of the selected marker.
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?	Places the selected marker at the next lowest peak on the trace.
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?	This command places the specified marker on a specified trace. The query form returns the trace on which the marker resides.
CALCulate:NOISe:TEMPerature:MARKer<x>:TRACe	Sets or queries which trace the specified marker is attached to.
CALCulate:NOISe:TEMPerature:MARKer<x>:X?	Queries the frequency of the specified marker.
CALCulate:NOISe:TEMPerature:MARKer<x>:Y?	Queries the temperature (Kelvin) at the specified marker.
CALCulate:NOISe:YFACtor:MARKer<x>:DELTA:X?	Queries the frequency difference between MR (reference marker) and the selected marker.
CALCulate:NOISe:YFACtor:MARKer<x>:DELTA:Y?	Queries the power difference value between MR (reference marker) and the selected marker.

Table continued...



Command	Description
CALCulate:NOISe:YFACTOR:MARKer<x>:MAXimum	Places the selected marker at the maximum value on the peak of the trace.
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LEFT?	Places the selected marker at the next peak to the left on the trace.
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LOWer?	Places the selected marker at the next lowest peak on the trace.
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:RIGHT?	Places the specified marker on a specified trace. The query form returns the trace on which the marker resides.
CALCulate:NOISe:YFACTOR:MARKer<x>:TRACe	Sets or queries which trace the specified marker is attached to.
CALCulate:NOISe:YFACTOR:MARKer<x>:X?	Queries the frequency of the specified marker.
CALCulate:NOISe:YFACTOR:MARKer<x>:Y?	This command queries the vertical position (dB) of the specified marker.
<b>CALCulate:P25 subcommand group</b>	<b>P25 analysis</b>
CALCulate:P25:CONStE:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:P25:CONStE:MARKer<x>:FDEVIation?	Queries the frequency deviation of the selected marker.
CALCulate:P25:CONStE:MARKer<x>:MAGNitude?	Queries the frequency deviation (for Freq Dev trace types) or magnitude (for IQ trace types) of the selected marker.
CALCulate:P25:CONStE:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace.
CALCulate:P25:CONStE:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:P25:CONStE:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:P25:CONStE:MARKer<x>:PHASe?	Queries the phase of the selected marker.
CALCulate:P25:CONStE:MARKer<x>:SYMBol?	Queries the readout symbol of the selected marker.
CALCulate:P25:CONStE:MARKer<x>:Trace	Places the selected marker on the Demodulated I and Q versus Time trace. The query returns the name of the trace on which the marker resides.
CALCulate:P25:CONStE:MARKer<x>:VALue?	Queries the readout value of the specified marker.
CALCulate:P25:CONStE:MARKer<x>:X	Sets or queries the horizontal position of the selected marker.
CALCulate:P25:EDIagram:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:P25:EDIagram:MARKer<x>:DELTA:Y?	Returns the delta marker time for the selected marker.
CALCulate:P25:EDIagram:MARKer<x>:MAXimum	Moves the selected marker to the highest peak.
CALCulate:P25:EDIagram:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next highest peak.
CALCulate:P25:EDIagram:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left .
CALCulate:P25:EDIagram:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next lowest peak.
CALCulate:P25:EDIagram:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right .
CALCulate:P25:EDIagram:MARKer<x>:TRACe	Sets the selected marker on the I or Q trace, or queries the trace that the specified marker is on.
CALCulate:P25:EDIagram:MARKer<x>:X	Sets or queries the horizontal position of the selected marker.
CALCulate:P25:EDIagram:MARKer<x>:Y	Sets or queries the vertical position of the selected marker.
CALCulate:P25:PVTIME:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time (seconds) for the selected marker.
CALCulate:P25:PVTIME:MARKer<x>:DELTA:Y[:TIME]?	Returns the Delta marker amplitude (dB) for the selected marker.

Table continued...

Command	Description
CALCulate:P25:PVTime:MARKer<x>:Maximum	Moves the selected marker to the highest peak on the trace.
CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude on the trace.
CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the trace.
CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:P25:PVTime:MARKer<x>:X	Sets or queries the horizontal position of the selected marker.
CALCulate:P25:PVTime:MARKer<x>:Y	Sets or queries the vertical position of the selected marker.
<b>CALCulate: Marker subgroup</b>	<b>Marker position and values</b>
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:DENSity?	Returns the power density of the specified marker.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:IDENSity	Returns the integrated power density reading between the MR (reference marker) and the selected marker.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:INTEgrated?	Returns the integrated power reading between the MR (reference marker) and the selected marker.
CALCulate:MARKer:AOFF	Turns off all markers.
CALCulate:MARKer:DELete	Deletes the last marker added.
CALCulate:MARKer:DENSity:EXCursion	Sets or queries the minimum excursion of DPX signal density.
CALCulate:MARKer:DENSity:SMOothing	Sets or queries the number of pixels squared for smoothing the density.
CALCulate:MARKer:DENSity:THReshold	Sets or queries the threshold of DPX signal density to detect peaks.
CALCulate:MARKer:DRAG:SEARch:STATe	Enables or disables peak searching when dragging a marker in a plot.
CALCulate:MARKer:MODE	Sets or queries the marker mode.
CALCulate:MARKer:NOISe:MODE	Sets or queries the marker noise mode.
CALCulate:MARKer:PEAK:EXCursion	Sets or queries the minimum excursion level.
CALCulate:MARKer:PEAK:THReshold	Sets or queries the threshold level to detect peaks.
<b>CALCulate:SEARch subgroup</b>	<b>Calculate search commands</b>
CALCulate:SEARch:LIMit:FAIL?	Queries whether the waveform cuts across the limit or not.
CALCulate:SEARch:LIMit:MATCH:BEEP[:STATe]	Sets or queries whether to beep when a match occurs.
CALCulate:SEARch:LIMit:MATCH:SACQuire[:STATe]	Sets or queries whether to stop acquiring data on match.
CALCulate:SEARch:LIMit:MATCH:SDATa[:STATe]	Sets or queries whether to save the acquisition data automatically.
CALCulate:SEARch:LIMit:MATCH:SPICture[:STATe]	Sets or queries whether to save the whole screen automatically.
CALCulate:SEARch:LIMit:MATCH:STRace[:STATe]	Sets or queries whether to save the spectrum trace automatically.
CALCulate:SEARch:LIMit:OPERation	Sets or queries the limit operation in the search function.
CALCulate:SEARch:LIMit:OPERation:FEED	Sets or queries the data flow to be fed in the search operation.
CALCulate:SEARch:LIMit:OPERation:MASK:LOAD	Loads the limit mask from a specified file for the search operation.

Table continued...

Command	Description
CALCulate:SEARCH:LIMit:OPERation:MASK:STORE	Stores the limit mask to a specified file for the search operation.
CALCulate:[:SEARCH]:TEST]:LIMit[:STATE]:MATCh:FILE:LOCation	Sets or queries the limit value in the search operation.
CALCulate:SEARCH:LIMit:REPort:DATA?	Returns the frequency range(s) that satisfy the search condition.
CALCulate:SEARCH:LIMit:REPort:POINts?	Returns the number of range(s) that satisfy the search condition.
CALCulate:SEARCH:LIMit:STATe	Sets or queries whether to enable or disable the search function.
<b>CALCulate:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
CALCulate:ACPower:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:ACPower:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:ACPower:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:ACPower:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:ACPower:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:ACPower:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:ACPower:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:{AM FM PM} subgroup (Option 21 only)</b>	<b>AM/FM/PM measurement</b>
CALCulate:{AM FM PM}:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:{AM FM PM}:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:{AM FM PM}:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:{AM FM PM}:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:{AM FM PM}:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:{AM FM PM}:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:{AM FM PM}:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:{AM FM PM}:MARKer<x>:X	Sets or queries the horizontal position (time) of the marker.
CALCulate:{AM FM PM}:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:AUDio subgroup</b>	<b>Audio measurements</b>
CALCulate:AUDio:HARMonic:HNUMber	Sets or queries the number of harmonics.
CALCulate:AUDio:HARMonic:NHNumber	Sets or queries the number of non-harmonics.
CALCulate:AUDio:NHARmonic:EXCursion	Sets or queries the non-harmonic excursion value.
CALCulate:AUDio:NHARmonic:IGNore	Sets or queries the ignore region of the non-harmonics.
CALCulate:AUDio:NHARmonic:THREshold	Sets or queries the non-harmonic threshold value.
CALCulate:AUDio:SPECTrum:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker on the spectrum trace.
CALCulate:AUDio:SPECTrum:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker on the spectrum trace.
CALCulate:AUDio:SPECTrum:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the spectrum trace.
CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

Table continued...

Command	Description
CALCulate:AUDio:SPECtrum:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the spectrum trace.
CALCulate:AUDio:SPECtrum:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the spectrum trace.
CALCulate:AUDio:SPECtrum:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the spectrum trace.
CALCulate:AUDio:SPECtrum:MARKer<x>:TRACe	Sets or queries the trace on which the specified marker is placed in the spectrum measurement.
CALCulate:AUDio:SPECtrum:MARKer<x>:X	Sets or queries the horizontal position of the selected marker on the spectrum trace.
CALCulate:AUDio:SPECtrum:MARKer<x>:Y?	Queries the marker amplitude of the selected marker on the spectrum trace.
<b>CALCulate:AVTime subgroup</b>	<b>Frequency versus Time measurement</b>
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:TRACe	Sets or queries the trace on which the marker is placed.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker on the Constellation trace.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:FDEVIation?	Queries the frequency deviation of the selected marker in the Constellation measurement for an FSK modulated signal.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAGNitude?	Queries the magnitude readout of the selected marker in the Constellation measurement.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAXimum	Positions the selected marker at the symbol in the center of the time record.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the selected marker in the time domain to the next lower symbol number, relative to the previous marker position.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker in the time domain to the next higher symbol number, relative to the previous marker position.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PHASe?	Queries the phase readout of the selected marker in the Constellation measurement.
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:SYMBOL?	Queries the symbol readout of the selected marker in the Constellation measurement.

Table continued...

Command	Description
CALCulate:CONStell[:MEASview<y>]:MARKer<x>:VALue?	Queries the value readout of the selected marker in the Constellation measurement.
CALCulate:CONStell[:MEASview<y>]:MARKer<x>:X	Sets or queries the time position of the selected marker on the Constellation trace.
<b>CALCulate:DIQVtime subgroup (Option 21 only)</b>	<b>Demodulated I&amp;Q versus Time measurement</b>
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:TRACe	Places the selected marker on the Demodulated I&Q versus Time trace.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:X[:TIME]	Sets or queries the horizontal position (time) of the selected marker.
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:DPX subgroup</b>	<b>DPX spectrum measurement</b>
CALCulate:DPX:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:DPX:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:DPX:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:DPX:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:DPX:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:DPX:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:DPX:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:DPX:MARKer<x>:TRACe	Sets or queries the trace to attach the marker to.
CALCulate:DPX:MARKer<x>:X:AMPLitude	Sets or queries the amplitude position of the marker.
CALCulate:DPX:MARKer<x>:X[:FREQuency]	Sets or queries the frequency position of the marker.
CALCulate:DPX:MARKer<x>:X:PHASe	Sets or queries the phase position of the selected marker.
CALCulate:DPX:MARKer<x>:X:TIME	Sets or queries the time position of the selected marker.
CALCulate:DPX:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:EDiagram subgroup (Option 21 only)</b>	<b>Eye diagram</b>
CALCulate:EDiagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker on the eye diagram trace.
CALCulate:EDiagram[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker on the eye diagram trace.
CALCulate:EDiagram[:MEASview<y>]:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the eye diagram trace.

Table continued...

Command	Description
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude on the eye diagram trace.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the eye diagram trace.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the eye diagram trace.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the eye diagram trace.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:TRACe	Places the selected marker on the I or Q trace in the eye diagram display.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:X[:TIME]	Sets or queries the horizontal position (time) of the selected marker in the eye diagram measurement.
CALCulate:EDDiagram[:MEASview<y>]:MARKer<x>:Y?	Queries the vertical position of the selected marker in the eye diagram measurement.
<b>CALCulate:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker amplitude for the selected marker in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude on the EVM versus Time trace.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the EVM versus Time trace.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the selected marker in the EVM versus Time measurement.
CALCulate:EVM[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker in the EVM versus Time measurement.
<b>CALCulate:FDVTime subgroup (Option 21 only)</b>	<b>Frequency deviation versus Time measurement</b>
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker on the Frequency deviation versus Time trace.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker in the Frequency deviation versus Time measurement.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace in the Frequency deviation versus Time measurement.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.

Table continued...

Command	Description
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the Frequency deviation versus Time trace.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace in the Frequency deviation versus Time measurement.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:X[:TIME]	Sets or queries the horizontal position of the selected marker in the Frequency deviation versus Time measurement.
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker in the Frequency deviation versus Time measurement.
<b>CALCulate:{FSETtling PSETtling} subgroup</b>	<b>Frequency and Phase Settling measurements</b>
CALCulate:{FSETtling PSETtling}:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker in the Frequency and Phase Settling displays.
CALCulate:{FSETtling PSETtling}:MARKer<x>:DELTA:Y?	Returns the delta marker frequency for the selected marker in the Frequency and Phase Settling displays.
CALCulate:{FSETtling PSETtling}:MARKer<x>:MAXimum	Moves the selected marker to the highest peak on the trace in the Frequency or Phase Settling display.
CALCulate:{FSETtling PSETtling}:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude on the Frequency or Phase Settling trace.
CALCulate:{FSETtling PSETtling}:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace in the Frequency or Phase Settling display.
CALCulate:{FSETtling PSETtling}:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the Frequency or Phase Settling trace.
CALCulate:{FSETtling PSETtling}:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace in the Frequency or Phase Settling display.
CALCulate:{FSETtling PSETtling}:MARKer<x>:TRACe	Sets or queries the trace on which the specified marker is placed in the spectrum measurement
CALCulate:{FSETtling PSETtling}:MARKer<x>:X	Sets or queries the horizontal position of the selected marker in the Frequency and Phase Settling display.
CALCulate:{FSETtling PSETtling}:MARKer<x>:Y?	Queries the marker amplitude of the selected marker in the Frequency and Phase Settling display.
<b>CALCulate:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker frequency for the selected marker.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.

Table continued...

Command	Description
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:TRACe	Sets or queries the trace (I or Q) to place the marker.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:MCPower subgroup</b>	<b>MCPR measurement</b>
CALCulate:MCPower:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:MCPower:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:MCPower:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:MCPower:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:MCPower:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:MCPower:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:MCPower:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:MERRor:Subgroup</b>	<b>Magnitude error versus Time measurement</b>
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker amplitude for the selected marker.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
CALCulate:OBWidth:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:OBWidth:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:OBWidth:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:OBWidth:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:OBWidth:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:OBWidth:MARKer<x>[:SET]:CENTer	Sets the center frequency to the value at the marker position.

Table continued...



Command	Description
CALCulate:OBWidth:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:OBWidth:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:OFDM subgroup</b>	<b>Orthogonal frequency division multiplexing measurement</b>
CALCulate:OFDM:CONSt:MARKer<x>:FREQuency	Sets or queries the frequency position of the selected marker.
CALCulate:OFDM:CONSt:MARKer<x>:MAGNitude?	Queries the marker magnitude of the selected marker.
CALCulate:OFDM:CONSt:MARKer<x>:PHASe?	Queries the phase readout of the selected marker.
CALCulate:OFDM:CONSt:MARKer<x>:TIME	Sets or queries the X position of the selected marker.
CALCulate:OFDM:CONSt:MARKer<x>:TYPE?	Queries the data type of the selected marker.
CALCulate:OFDM:CONSt:MARKer<x>:VALue?	Queries the value of the specified marker.
CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:Y?	Returns the delta marker frequency for the selected marker.
CALCulate:OFDM:CRESPonse:MARKer<x>:MAXimum	Moves the selected marker to the highest peak.
CALCulate:OFDM:CRESPonse:MARKer<x>:PEAK:HIGHER	Moves the selected marker to the next peak higher in amplitude.
CALCulate:OFDM:CRESPonse:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left.
CALCulate:OFDM:CRESPonse:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
CALCulate:OFDM:CRESPonse:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right.
CALCulate:OFDM:CRESPonse:MARKer<x>:X	Sets or queries the horizontal position of the selected marker.
CALCulate:OFDM:CRESPonse:MARKer<x>:Y?	Returns the amplitude of the selected marker.
CALCulate:OFDM:FLATness:MARKer<x>:MAXimum	Positions the specified marker at the maximum point on the trace.
CALCulate:OFDM:FLATness:MARKer<x>:PEAK:HIGHER	Moves the specified marker to the next peak higher in amplitude on the trace.
CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LEFT	Moves the specified marker to the next peak to the left on the trace.
CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude on the trace.
CALCulate:OFDM:FLATness:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:OFDM:FLATness:MARKer<x>:X	Sets or queries the horizontal value at the selected marker position on the points trace.
CALCulate:OFDM:FLATness:MARKer<x>:Y?	Returns the value of the amplitude (vertical position) at the selected marker position.
CALCulate:OFDM:TABLE:MARKer<x>:FREQuency	Sets or queries the frequency position of the selected marker.
CALCulate:OFDM:TABLE:MARKer<x>:TIME	Sets or queries the marker time.
CALCulate:OFDM:TABLE:MARKer<x>:VALue?	Queries the value readout of the selected marker.
<b>CALCulate:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker amplitude for the selected marker in the Phase error versus Time measurement.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker phase for the selected marker.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHER	Moves the marker to the next peak higher in amplitude.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.

Table continued...

Command	Description
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker phase for the selected marker.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:PULSe subgroup</b>	<b>Pulsed RF measurements</b>
CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:X	Sets or queries the frequency in Hz for the X cursor.
CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:Y?	Returns the frequency level in dB for the Y cursor.
CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:X	Sets or queries the histogram indicator X value.
CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:Y?	Queries the histogram indicator Y value.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the specified marker.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the specified marker.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the statistics trace.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the statistics trace.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.

Table continued...

Command	Description
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the pulse trace.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the pulse trace.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the pulse trace.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:SEARch subgroup</b>	<b>Spectrogram measurement</b>
CALCulate:SEARch:LIMit:FAIL?	Queries whether the waveform cuts across the limit in the search operation.
CALCulate:SEARch:LIMit:MATCh:BEEP[:STATe]	Sets or queries whether or not to emit a beep on match during run or replay in the search operation.
CALCulate:SEARch:LIMit:MATCh:SACQuire[:STATe]	Sets or queries whether or not to stop acquiring data on match during run or replay in the search operation.
CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]	Sets or queries whether or not to save automatically (AutoSave) acquisition data on match during run in the search operation.
CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe]	Sets or queries whether or not to save automatically (AutoSave) the whole screen on match during run in the search operation.
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]	Sets or queries whether or not to save automatically (AutoSave) the spectrum trace on match during run in the search operation.
CALCulate:SEARch:LIMit:OPERation	Sets or queries the search condition parameter.
CALCulate:SEARch:LIMit:OPERation:FEED	Sets or queries the data flow to be fed in the search operation.
CALCulate:SEARch:LIMit:OPERation:MASK:LOAD	Loads the specified limit mask file for the search operation.
CALCulate:SEARch:LIMit:OPERation:MASK:STORE	Saves the limit mask to a specified file.
CALCulate[:SEARch]:TEST[:LIMit[:STATe]:MATCh:FILE:LOCation	Sets or queries the limit value (in dB) in the search operation.
CALCulate:SEARch:LIMit:REPort:DATA?	Queries the frequency range(s) that satisfy the search condition.
CALCulate:SEARch:LIMit:REPort:POINts?	Queries the number of frequency range(s) that satisfy the search condition.
CALCulate:SEARch:LIMit:STATe	Sets or queries the search function (enabled or disabled).
<b>CALCulate:SEM subgroup</b>	<b>Spectral Emissions Mask measurement</b>
CALCulate:SEM:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:SEM:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SEM:MARKer<x>:MAXimum	Moves the selected marker to the highest peak.
CALCulate:SEM:MARKer<x>:PEAK:HIGHer	Moves the selected marker to the next peak higher in amplitude.
CALCulate:SEM:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left.
CALCulate:SEM:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude.
Table continued...	

Command	Description
CALCulate:SEM:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right.
CALCulate:SEM:MARKer<x>[:SET]:CENTer	Moves the specified marker to the center frequency.
CALCulate:SEM:MARKer<x>:X	Sets or queries the horizontal position of the selected marker.
CALCulate:SEM:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>[CALCulate]:SGRam subgroup</b>	<b>Spectrogram measurements</b>
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X:FREQue ncy?	Returns the delta marker frequency for the selected marker.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X:FREQuency	Sets or queries the marker frequency.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X[:TIME]	Sets or queries the marker time.
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:SPECTrum subgroup</b>	<b>Spectrogram measurement</b>
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:TRACe	Sets or queries the trace on which the marker is placed.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:SPURious subgroup</b>	<b>Spurious measurement</b>
CALCulate:SPURious:MARKer<x>:DELTA:X?	Returns the delta marker frequency for the selected marker.
CALCulate:SPURious:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:SPURious:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:SPURious:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:SPURious:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:SPURious:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.

Table continued...

Command	Description
CALCulate:SPURious:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:SPURious:MARKer<x>[:SET]:CENTer	Sets the center frequency to the marker frequency.
CALCulate:SPURious:MARKer<x>:X	Sets or queries the horizontal position of the marker.
CALCulate:SPURious:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:TDiagram subgroup (Option 21 only)</b>	<b>Trellis diagram</b>
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?	Returns the delta marker time for the selected marker.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude on the Trellis diagram trace.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:X[:TIME]	Sets or queries the horizontal position (time) of the marker.
CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:Y?	Queries the vertical position of the selected marker in the Trellis diagram measurement.
<b>CALCulate:TOverview subgroup</b>	<b>Time overview</b>
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak to the left on the trace.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:X	Sets or queries the horizontal position of the marker
CALCulate:TOverview[:MEASview<y>]:MARKer<x>:Y?	Queries the marker amplitude of the selected marker.
<b>CALCulate:WLAN subgroup</b>	<b>Wireless LAN measurement</b>
CALCulate:WLAN:CONSt:MARKer<x>:FREQUency	Sets or queries the frequency value of the marker.
CALCulate:WLAN:CONSt:MARKer<x>:MAGNitude?	Returns the magnitude value at the indicated marker position.
CALCulate:WLAN:CONSt:MARKer<x>:PHASe?	Returns the phase value at the indicated marker position.
CALCulate:WLAN:CONSt:MARKer<x>:TIME	Sets the time offset value at the selected marker position on the points trace.
CALCulate:WLAN:CONSt:MARKer<x>:TYPE?	Returns the data type of the subcarrier for data at the indicated marker position.
CALCulate:WLAN:CONSt:MARKer<x>:VALue?	Queries the value readout at the selected marker position on the points trace.
CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:X?	Queries the delta Subcarrier or frequency value at the selected marker position.

Table continued...

Command	Description
CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:Y?	Queries the vertical delta value for the points trace at the selected marker position.
CALCulate:WLAN:CRESPonse:MARKer<x>:MAXimum	Moves the marker to the highest peak on the trace.
CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:HIGHer	Moves the marker to the next peak higher in amplitude.
CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:LEFT	Moves the marker to the next peak on the left on the trace.
CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right on the trace.
CALCulate:WLAN:CRESPonse:MARKer<x>:X	Sets or queries the value of the horizontal position for the selected marker.
CALCulate:WLAN:CRESPonse:MARKer<x>:Y?	Returns the value of the amplitude at the selected marker position.
CALCulate:WLAN:EVM:MARKer<x>:AVERAge:SUBCarrier?	Queries the EVM value at the selected marker position on the average trace.
CALCulate:WLAN:EVM:MARKer<x>:AVERAge:SYMBOL?	Queries the EVM value at the selected marker position on the average trace.
CALCulate:WLAN:EVM:MARKer<x>:DELTA:SUBCarrier?	Queries the delta Subcarrier or frequency value at the selected marker position.
CALCulate:WLAN:EVM:MARKer<x>:DELTA:SYMBOL?	Queries the delta time value in seconds or symbols at the selected marker position.
CALCulate:WLAN:EVM:MARKer<x>:DELTA:Y?	Queries the vertical delta value for the points trace at the selected marker position.
CALCulate:WLAN:EVM:MARKer<x>:FREQUency	Sets or queries the frequency value of the marker.
CALCulate:WLAN:EVM:MARKer<x>:TIME	Sets the vertical marker value of the points trace.
CALCulate:WLAN:EVM:MARKer<x>:VALue?	Queries the EVM value at the selected marker position.
CALCulate:WLAN:FLATness:MARKer<x>:MAXimum	Positions the specified marker at the maximum point on the trace.
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:HIGHer	Moves the specified marker to the next peak higher in amplitude.
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LEFT	Moves the specified marker to the next peak to the left.
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right.
CALCulate:WLAN:FLATness:MARKer<x>:X	Sets or queries the horizontal position for the selected marker.
CALCulate:WLAN:FLATness:MARKer<x>:Y?	Returns the value of the amplitude (vertical position) at the selected marker position.
CALCulate:WLAN:MERRor:MARKer<x>:AVERAge:SUBCarrier?	Queries the value readout at the selected marker position on the average trace.
CALCulate:WLAN:MERRor:MARKer<x>:AVERAge:SYMBOL?	Queries the value readout at the selected marker position on the average trace.
CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SUBCarrier?	Queries the delta Subcarrier or frequency value at the selected marker position.
CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SYMBOL?	Queries the delta time value in seconds or symbols at the selected marker position.
CALCulate:WLAN:MERRor:MARKer<x>:DELTA:Y?	Queries the vertical delta value for the points trace at the selected marker position.

Table continued...

Command	Description
CALCulate:WLAN:MERRor:MARKer<x>:FREQUency	Sets or queries the frequency value of the marker.
CALCulate:WLAN:MERRor:MARKer<x>:TIME	Sets the time offset value at the selected marker position.
CALCulate:WLAN:MERRor:MARKer<x>:VALue?	Queries the value readout at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:AVERAge:SUBCarrier?	Queries the Phase Error value at the selected marker position on the average trace.
CALCulate:WLAN:PERRor:MARKer<x>:AVERAge:SYMBOL?	Queries the Phase Error value at the selected marker position on the average trace.
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SUBCarrier?	Queries the delta Subcarrier or frequency value at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SYMBOL?	Queries the delta time value in seconds or symbols at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:Y?	Queries the vertical delta value for the points trace at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:FREQUency	Sets or queries the frequency value at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:TIME	Sets the time value (offset) at the selected marker position.
CALCulate:WLAN:PERRor:MARKer<x>:VALue?	Queries the Phase Error value at the selected marker position.
CALCulate:WLAN:PVTime:MARKer<x>:DELTA:X?	Returns the delta time value at the selected marker position.
CALCulate:WLAN:PVTime:MARKer<x>:DELTA:Y?	Queries the vertical delta value for the points trace at the selected marker position.
CALCulate:WLAN:PVTime:MARKer<x>:MAXimum	Positions the specified marker at the maximum point.
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:HIGHer	Moves the specified marker to the next peak higher in amplitude.
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LEFT	Moves the specified marker to the next peak on the left.
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LOWer	Moves the marker to the next peak lower in amplitude.
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:RIGHT	Moves the marker to the next peak to the right.
CALCulate:WLAN:PVTime:MARKer<x>:X	Sets or queries the value of the horizontal position for the selected marker
CALCulate:WLAN:PVTime:MARKer<x>:Y?	Returns the amplitude at the selected marker position.
CALCulate:WLAN:TABLE:MARKer<x>:FREQUency	Assigns or queries the frequency value of the marker.
CALCulate:WLAN:TABLE:MARKer<x>:TIME	Sets or queries the vertical value at the selected marker position.
CALCulate:WLAN:TABLE:MARKer<x>:VALue?	Queries the decoded symbol value at the selected marker position.
<b>CALCulate:LTE subgroup</b>	
CALCulate:LTE:ACLR:MARKer<x>:PEAK:HIGHer	Moves the specified marker to the next peak higher in amplitude on the trace.
CALCulate:LTE:ACLR:MARKer<x>:DELTA:X?	Returns the frequency of the specified delta marker on the trace.
CALCulate:LTE:ACLR:MARKer<x>:DELTA:Y?	Returns the amplitude for the specified delta marker on the trace.
CALCulate:LTE:ACLR:MARKer<x>:MAXimum	Positions the specified marker at the highest peak on the trace.
CALCulate:LTE:ACLR:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:LTE:ACLR:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the trace.

Table continued...

Command	Description
CALCulate:LTE:ACLR:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:LTE:ACLR:MARKer<x>:SET:CENTer	Moves the specified marker to the center frequency.
CALCulate:LTE:ACLR:MARKer<x>:X	Sets or queries the horizontal position of the selected marker on the trace.
CALCulate:LTE:ACLR:MARKer<x>:Y?	Queries the amplitude of the selected marker on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:X?	Returns the frequency of the specified delta marker on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:Y?	Returns the amplitude for the specified delta marker on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:MAXimum	Positions the marker at the highest peak on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:X	Sets or queries the horizontal position of the selected marker on the trace.
CALCulate:LTE:CHSPectrum:MARKer<x>:Y?	Queries the amplitude of the selected marker on the trace.
CALCulate:LTE:CONSte:MARKer<x>:FREQUency	Sets or returns the subcarrier number of the LTE symbol.
CALCulate:LTE:CONSte:MARKer<x>:MAGNitude?	Queries the marker magnitude of the selected marker.
CALCulate:LTE:CONSte:MARKer<x>:PHASe?	Queries the phase readout of the selected marker.
CALCulate:LTE:CONSte:MARKer<x>:TIME	Sets or queries the X position (Symbol Value) of the selected marker.
CALCulate:LTE:CONSte:MARKer<x>:TYPE?	Queries the data type of the selected marker as PSS or SSS.
CALCulate:LTE:PVTIme:MARKer<x>:PEAK:HIGHer	Moves the specified marker to the next peak higher in amplitude on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:DELTA:X?	Returns the frequency of the specified delta marker on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:DELTA:Y?	Returns the amplitude for the specified delta marker on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:MAXimum	Positions the specified marker at the highest peak on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:PEAK:LEFT	Moves the selected marker to the next peak to the left on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:PEAK:LOWer	Moves the selected marker to the next peak lower in amplitude on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:PEAK:RIGHT	Moves the selected marker to the next peak to the right on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:X	Sets or queries the horizontal position of the selected marker on the trace.
CALCulate:LTE:PVTIme:MARKer<x>:Y?	Queries the amplitude of the selected marker on the trace.
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:HIGHer?	Places the selected marker at the next highest peak on the trace.
CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:X	Sets or queries the horizontal position of the indicator on the trace.
CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:Y?	Queries the vertical position value of the indicator on the trace.

Table continued...



Command	Description
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:X?	Returns the delta marker time for the selected marker.
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:Y?	Returns the delta marker amplitude for the selected marker.
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:MAXimum	Moves the selected marker to the highest peak on the trace.
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:X	Sets or queries the horizontal position of the selected marker.
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:Y	Sets or queries the marker amplitude of the selected marker.
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:X?	Returns the delta marker time for the selected marker.

## Marker mnemonics

Up to five markers can be used. In commands, these are named MARKer<x>, where <x> can be 0, 1, 2, 3, or 4 as shown in the following table.

**Table 14: Marker mnemonics**

Mnemonic	Description
MARKer0	Reference marker (MR)
MARKer1	Marker 1 (M1)
MARKer2	Marker 2 (M2)
MARKer3	Marker 3 (M3)
MARKer4	Marker 4 (M4)

**Note:**



*If you omit the numeric suffix, the marker control defaults to Marker 1.*

*Before operating the marker, you have to enable it using the CALCulate basic commands.*

*If you attempt to use a marker other than above in a CALCulate command, the suffix error (error code -130) will occur.*

## Calibration commands

Use the CALibration commands to control the external correction. See the Index for commands not listed in the following table.

Not all of these commands apply to SignalVu. See the individual command descriptions for details.

**Table 15: Calibration commands**

Command	Description
CALibration:ABORT	Aborts any actions related to the alignments in progress.
CALibration:TXGain:ABORT	Cancels a calibration for the Transmission Gain measurement.
CALibration:TXGain[:ALL]	Performs a calibration using the current settings for the Transmission Gain measurement.

Table continued...

Command	Description
CALibration:TXGain:FINish?	Queries whether or not the calibration is completed for the Transmission Gain measurement.
CALibration:TXGain:STATe?	Queries the calibration state for the Transmission Gain measurement.
CALibration:AUTO	Sets or queries whether or not to run alignments automatically.
INPut:CORRection:EXTernal:EDIT<x>:LABel	Sets or queries the name of the external loss table.
CALibration:CORRection:EXTernal:EDIT<x>:NEW	Creates a new external loss table.
INPut:CORRection:EXTernal:EDIT<x>:STATe	Sets or queries whether to enable or disable the external loss table.
INPut:CORRection:EXTernal:GAIN[:MAGNitude]	Sets or queries the external gain value.
CALibration:CORRection:EXTernal:PROBe:CONNect?	Sets or queries whether to enable or disable the external gain value.
CALibration:CORRection:EXTernal:PROBe:CONNect?	Queries whether the external probe is connected to the analyzer or not.
CALibration:CORRection:EXTernal:PROBe[:MAGNitude]?	Queries the external probe attenuation.
CALibration:CORRection:EXTernal:PROBe:STATe	Determines whether to correct data for the external probe attenuation.
CALibration:CORRection:EXTernal:TYPE	Selects data type to use when applying the external loss table.

## Display commands

Use the DISPlay commands to control the display of measurement waveforms and results on the screen. See the Index for commands not listed in the following table.

Not all of these commands apply to SignalVu. See the individual command descriptions for details.

**Table 16: Display commands**

Command	Description
<b>DISPlay TXGain subgroup</b>	<b>Transmission Gain measurements</b>
DISPlay:TG:MEASview<y>:DELeTe TXGain	Removes the Transmission Gain display icon from the Selected displays filed of the Selected Displays window.
DISPlay:TG:MEASview<y>:NEW TXGain	Opens a new Transmission Gain measurement display window.
DISPlay:TG:MEASview<y>:SELeCt TXGain	Selects the Transmission Gain display icon in the Select Display window and queries if the display is selected or not.
DISPlay:TXGain:MARKer:SHOW:STATe	Displays or hides the marker readout, but not the marker itself, on the graph area.
DISPlay:TXGain[:SCALe]:AUTO	Automatically rescales the horizontal and vertical scale.
DISPlay:TXGain:WINDow:TRACe:GRATICule:GRID:STATe	Displays or hides the graticule grid in the display. Queries whether the graticule grid is hidden or showing.
DISPlay:TXGain:WINDow:TRACe:POINts:SHOW:STATe	Sets to mark each measurement point on the trace.
DISPlay:TXGain:X[:SCALe]:LOG:STATe	Sets or queries the display to show the frequency axis in a logarithmic scale.
DISPlay:TXGain:X[:SCALe]:START	Sets the Zoom Start frequency.

Table continued...

Command	Description
DISPlay:TXGain:X[:SCALe]:AUTO	Resets the scale of the horizontal axis to contain the complete trace.
DISPlay:TXGain:X[:SCALe]:STOP	Sets the Zoom Stop frequency.
DISPlay:TXGain:Y[:SCALe]:BOTTom	Sets or queries the bottom of the vertical position of the display.
DISPlay:TXGain:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division) of the graph.
DISPlay:TXGain:Y[:SCALe]:POSition	Sets or queries the vertical position of the trace.
DISPlay:TXGain:Y[:SCALe]:TOP	Sets or queries the top of the vertical scale.
DISPlay:TXGain:Y[:SCALe]:AUTO	Automatically scales the plot vertically.
DISPlay:TXGain:Y[:SCALe]	Sets or queries the vertical scale value in dB.
<b>DISPlay Noise subgroup</b>	<b>Noise Figure and Gain measurements</b>
DISPlay:WINDow:ACTive:MEASurement?	Queries the active measurement views.
<b>DISPlay Bluetooth subgroup</b>	<b>Bluetooth measurements</b>
DISPlay:BIBEmissions:MARKer<x>:SHOW:STATe	Sets or queries to show or hide the readout for the selected marker.
DISPlay:BIBEmissions:PLEVel:SHOW:STATe	Sets or queries to show or hide the power levels
DISPlay:BIBEmissions:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:BIBEmissions:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule hidden or showing state.
DISPlay:BIBEmissions:X[:SCALe]	Sets or queries the horizontal range of the graph.
DISPlay:BIBEmissions:X[:SCALe]:AUTO	Rescales the horizontal axis automatically to fit the waveform to the screen in the Bluetooth InBand Emission view.
DISPlay:BIBEmissions:X[:SCALe]:OFFSet	Sets or queries the center frequency.
DISPlay:BIBEmissions:Y[:SCALe]	Sets or queries the vertical range of the graph.
DISPlay:BIBEmissions:Y[:SCALe]:AUTO	Rescales the vertical axis automatically to fit the waveform to the screen in the Bluetooth InBand Emission view.
DISPlay:BIBEmissions:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:BLUEtooth:CONStE:TRACe:GRATICule:GRID:STATe	Sets or queries the display state of graticule grid.
DISPlay:BLUEtooth:CONStE:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show or hide the graticule grid on the screen in the display.
DISPlay:BLUEtooth:EDIagram:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries graticule grid hidden or showing state in the Bluetooth eye diagram display.
DISPlay:BLUEtooth:EDIagram:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:BLUEtooth:EDIagram:Y[:SCALe]:AUTO	Sets the vertical scale automatically to fit the waveform in the Bluetooth eye diagram.
DISPlay:BLUEtooth:EDIagram:Y[:SCALe]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis).
DISPlay:BLUEtooth:FDVTime:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries graticule grid hidden or showing state in the Freq Dev vs Time display.
DISPlay:BLUEtooth:FDVTime:Y[:SCALe]:AUTO	Rescales the vertical scale automatically to fit the waveform to the screen.
DISPlay:BLUEtooth:FDVTime:Y[:SCALe]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis).
DISPlay:BLUEtooth:MEASview<y>:DELeTe	Deletes the specified Bluetooth analysis view.

Table continued...

Command	Description
DISPlay:BLUetooth:MEASview<y>:NEW	Displays a new measurement view for Bluetooth measurements.
DISPlay:BLUetooth:MEASview<y>:SElect	Selects a measurement view on the screen in the Bluetooth measurements. The query command returns the currently selected view.
DISPlay:BOBW:SElecteD:BA NDwidth	Sets or queries the bandwidth to measure in the Bluetooth 20 dB BW view.
DISPlay:DPX:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule grid on the DPX Spectrum display..
DISPlay:NOISe:FIGure[:SCALe]:AUTO	Automatically scales the graph.
DISPlay:NOISe:FIGure:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule state to on (showing) or off (hidden).
DISPlay:NOISe:FIGure:WINDow:TRACe:LEGenD:STATe	Sets or queries the trace legend state to on (showing) or off (hidden).
DISPlay:NOISe:FIGure:WINDow:TRACe:MEASpoints:STATe	Sets or queries the measurement points state to on (showing) or off (hidden).
DISPlay:NOISe:FIGure:X:AUTO	Automatically sets the horizontal scale of the graph.
DISPlay:NOISe:FIGure:X:OFFSet	Sets or queries the center frequency (offset).
DISPlay:NOISe:FIGure:X[:SCALe]	Sets or queries the horizontal scale of the display.
DISPlay:NOISe:FIGure:Y:AUTO	Automatically sets the vertical scale of the graph.
DISPlay:NOISe:FIGure:Y:AUTO:STATe	Sets the state of the automatic vertical scale and position of the graph to on or off. The query form of the command returns the current state.
DISPlay:NOISe:FIGure:Y:PDIVision	Sets or queries the distance between graticule lines on the display.
DISPlay:NOISe:FIGure:Y:POSition	Sets or queries the vertical position of the display.
DISPlay:NOISe:FIGure:Y:SCALe	Sets or queries the vertical scale of the display.
DISPlay:NOISe:FIGure:Y:SCALe:LINear:STATe	Sets or queries whether the linear units are shown (on) or not shown (off) on the display.
DISPlay:NOISe:GAIN[:SCALe]:AUTO	Automatically scales the graph.
DISPlay:NOISe:GAIN:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule state to on (showing) or off (hidden).
DISPlay:NOISe:GAIN:WINDow:TRACe:LEGenD:STATe	Sets or queries the trace legend state to on (showing) or off (hidden).
DISPlay:NOISe:GAIN:WINDow:TRACe:MEASPoints:STATe	Sets or queries the measurement points state to on (showing) or off (hidden).
DISPlay:NOISe:GAIN:X:AUTO	Automatically sets the horizontal scale of the display.
DISPlay:NOISe:GAIN:X:OFFSet	Sets or queries the center frequency (offset).
DISPlay:NOISe:GAIN:X:SCALe	Sets or queries the horizontal scale of the display.
DISPlay:NOISe:GAIN:Y:AUTO	Automatically sets the vertical scale of the graph.
DISPlay:NOISe:GAIN:Y:AUTO:STATe	Sets or queries the state of the automatic vertical scale and position of the graph to on or off.
DISPlay:NOISe:GAIN:Y:PDIVision	Sets or queries the distance between graticule lines on the display.
DISPlay:NOISe:GAIN:Y:POSition	Sets or queries the vertical position of the display.

Table continued...

Command	Description
DISPlay:NOISe:GAIN:Y:SCALE	Sets or queries the vertical scale of the display.
DISPlay:NOISe:MEASview<y>:DELeTe	Deletes the selected Noise and Gain measurement view.
DISPlay:NOISe:MEASview<y>:NEW	Displays a new Noise and Gain measurement view.
DISPlay:NOISe:MEASview<y>:SELeCt	Selects a noise measurement view. The query command returns the currently selected view.
DISPlay:NOISe:POWer:LINEar:STATe	Sets or queries whether the linear units are shown (on) or not shown (off) on the Noise Table display.
DISPlay:NOISe:TEMPerature[:SCALE]:AUTO	Automatically scales the graph.
DISPlay:NOISe:TEMPerature:WINDow:TRACe:GRATicule:GRID:STATe	Sets or queries the graticule state to on (showing) or off (hidden).
DISPlay:NOISe:TEMPerature:WINDow:TRACe:LEGend:STATe	Sets or queries the trace legend state to on (showing) or off (hidden).
DISPlay:NOISe:TEMPerature:WINDow:TRACe:MEASPoints:STATe	Sets or queries the measurement points state to on (showing) or off (hidden).
DISPlay:NOISe:TEMPerature:X:AUTO	Automatically sets the horizontal scale of the graph.
DISPlay:NOISe:TEMPerature:X:OFFSet	Sets or queries the center frequency (offset) value.
DISPlay:NOISe:TEMPerature:X[:SCALE]	Sets or queries the horizontal scale of the graph.
DISPlay:NOISe:TEMPerature:X:START	Sets or queries the Start Frequency value.
DISPlay:NOISe:TEMPerature:X:STOP	Sets or queries the Stop Frequency value.
DISPlay:NOISe:TEMPerature:Y:AUTO	Automatically sets the vertical scale of the graph.
DISPlay:NOISe:TEMPerature:Y:AUTO:STATe	Sets or queries the state of the automatic vertical scale and position of the graph to on or off.
DISPlay:NOISe:TEMPerature:Y:PDIVision	Sets or queries the distance between graticule lines on the display.
DISPlay:NOISe:TEMPerature:Y:POSition	Sets or queries the vertical position of the graph.
DISPlay:NOISe:TEMPerature:Y:SCALE	Sets or queries the vertical scale of the graph.
DISPlay:NOISe:UNCERTainty:RESult:GAIN?	Queries the gain computed uncertainty value.
DISPlay:NOISe:UNCERTainty:RESult:NFIGure?	Queries the noise figure computed uncertainty value.
DISPlay:NOISe:YFACtor[:SCALE]:AUTO	Automatically scales the Y Factor graph.
DISPlay:NOISe:YFACtor:WINDow:TRACe:GRATicule:GRID:STATe	Sets or queries the graticule state to on (showing) or off (hidden).
DISPlay:NOISe:YFACtor:WINDow:TRACe:LEGend:STATe	Sets or queries the trace legend state to on (showing) or off (hidden).
DISPlay:NOISe:YFACtor:WINDow:TRACe:MEASPoints:STATe	Sets or queries the measurement points state to on (showing) or off (hidden).
DISPlay:NOISe:YFACtor:X:AUTO	Automatically sets the horizontal scale.
DISPlay:NOISe:YFACtor:X:OFFSet	Sets or queries the center frequency (offset).
DISPlay:NOISe:YFACtor:X[:SCALE]	Sets or queries the horizontal scale.
DISPlay:NOISe:YFACtor:X:START	Sets or queries the Start Frequency value.
DISPlay:NOISe:YFACtor:X:STOP	Sets or queries the Stop Frequency value.
DISPlay:NOISe:YFACtor:Y:AUTO	Automatically sets the vertical scale of the graph.

Table continued...

Command	Description
DISPlay:NOISe:YFACTOR:Y:AUTO:STATE	Sets or queries the automatic vertical scale and position of the graph setting to on or off.
DISPlay:NOISe:YFACTOR:Y:PDIVision	Sets or queries the distance between graticule lines on the display in dB.
DISPlay:NOISe:YFACTOR:Y:POSition	Sets or queries the vertical position.
DISPlay:NOISe:YFACTOR:Y:SCALE	Sets or queries the vertical scale.
DISPlay:P25:CONSte:WINDow:TRACe:GRATICule:GRID:STATE	Sets or queries to show or hide the graticule grid on the screen.
DISPlay:P25:EDIagram:WINDow:TRACe:GRATICule:GRID:STATE	Sets or queries to show or hide the graticule grid on the screen.
DISPlay:P25:EDIagram:Y[:SCALE]	Sets or queries the vertical scale.
DISPlay:P25:EDIagram:Y[:SCALE]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen.
DISPlay:P25:EDIagram:Y[:SCALE]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis).
DISPlay:P25:MEASview:DELeTe	Deletes the specified P25 measurement display.
DISPlay:P25:MEASview:NEW	Displays a new P25 measurement view.
DISPlay:P25:MEASview:SELeCt	Selects a P25 measurement display. The query form returns the currently selected display.
DISPlay:P25:PVTime:BURSt:X[:SCALE]	Sets or queries the value of the scale (width) value, in seconds, when using horizontal Full Burst view.
DISPlay:P25:PVTime:BURSt:X[:SCALE]:AUTO	Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Full Burst view.
DISPlay:P25:PVTime:BURSt:X[:SCALE]:OFFSet	Sets or queries the starting time (position) value, in seconds, when using the horizontal Full Burst view.
DISPlay:P25:PVTime:BURSt:X[:SCALE]:RESet	Resets the starting time (position) and scale (width) values for the horizontal Full Burst view.
DISPlay:P25:PVTime:FALL:X[:SCALE]	Sets or queries the scale (width) value, in seconds, when using the horizontal Falling Edge view.
DISPlay:P25:PVTime:FALL:X[:SCALE]:AUTO	Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Falling Edge view.
DISPlay:P25:PVTime:FALL:X[:SCALE]:OFFSet	Sets or queries the value for the starting time (offset) of the graph, in seconds, when using the horizontal Falling Edge view.
DISPlay:P25:PVTime:FALL:X[:SCALE]:RESet	Resets the starting time (position) and scale (width) values for the horizontal Falling Edge view.
DISPlay:P25:PVTime:MARKer:SHOW:STATE	Shows or hides the marker readout.
DISPlay:P25:PVTime:RISE:X[:SCALE]	Sets or queries the scale (width) value, in seconds, for the P25 Power vs. Time display when using the horizontal Rising Edge view.
DISPlay:P25:PVTime:RISE:X[:SCALE]:AUTO	Sets the scale (width) value, in seconds, to automatic.
DISPlay:P25:PVTime:RISE:X[:SCALE]:OFFSet	Sets or queries the starting time value (offset), in seconds, for the P25 Power vs. Time display when using the horizontal Rising Edge view.
DISPlay:P25:PVTime:RISE:X[:SCALE]:RESet	Resets the starting time (position) and scale (width) values for the horizontal Rising Edge view.

Table continued...

Command	Description
DISPlay:P25:PVTime:WINDow:SElect:PLOT	Sets or queries which view to use: Full Burst, Rising Edge, or Falling Edge.
DISPlay:P25:PVTime:WINDow:TRACe:GRATicule:GRID:STATe	Sets or queries the graticule state to on (showing) or off (hidden).
DISPlay:P25:PVTime:Y[:SCALe]	Sets or queries the vertical scale value.
DISPlay:P25:PVTime:Y[:SCALe]:AUTO	Automatically selects the vertical scale and position values.
DISPlay:P25:PVTime:Y[:SCALe]:OFFSet	Sets or queries the value of the vertical offset (top edge of the vertical axis).
DISPlay:P25:PVTime:Y[:SCALe]:PDIVision	Sets or queries the value of the vertical scale for the P25 Power vs. Time display.
DISPlay:P25:RADix	Set or queries the symbol radix for the P25 Symbol Table display.
DISPlay:TOVerview[:MEASview<y>]:WINDow:NAVigator:STATe	Sets or queries whether the navigator view of the Time Overview display is on or off.
DISPlay:TOVerview[:MEASview<y>]:WINDow:TIME:MODE	Sets or queries the type of time analysis to be performed.
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:LEGenD:STATe	Sets or queries the trace legend state to on (showing) or off (hidden).
<b>DISPlay basic command subgroup</b>	<b>General window control</b>
DISPlay:WINDow:COLor:SCHEME	Sets or queries the color scheme for traces and background.
DISPlay:WINDow:OPTimized:MEASurement?	Queries the measurement views that are optimized.
<b>DISPlay:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
DISPlay:ACPower:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:ACPower:PLEVel:SHOW:STATe	Determines whether to show the power levels.
DISPlay:ACPower:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:ACPower:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:ACPower:X[:SCALe]	Sets or queries the horizontal range.
DISPlay:ACPower:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:ACPower:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:ACPower:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:ACPower:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:ACPower:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:ADEMod subgroup (Option 21 only)</b>	<b>General purpose analog demodulation measurements</b>
DISPlay:ADEMod:MEASview<y>:DELete	Deletes the measurement view.
DISPlay:ADEMod:MEASview<y>:NEW	Displays a new measurement view.
DISPlay:ADEMod:MEASview<y>:SElect	Sets or queries the measurement view.
<b>DISPlay:{AM FM PM} subgroup (Option 21 only)</b>	<b>AM/FM/PM measurement</b>
DISPlay:{AM FM PM}:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:{AM FM PM}:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:{AM FM PM}:X:RSCale	Rescales the horizontal axis automatically.
DISPlay:{AM FM PM}:X[:SCALe]:AUTO	Sets the horizontal scale automatically.

Table continued...

Command	Description
DISPlay:{AM FM PM}:X[:SCALe]:FULL	Sets or queries the horizontal scale.
DISPlay:{AM FM PM}:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:{AM FM PM}:Y:RSCale	Rescales the vertical axis automatically.
DISPlay:{AM FM PM}:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:{AM FM PM}:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:AUDio subgroup</b>	<b>Audio measurements</b>
DISPlay:AUDio:MEASview<y>:DELeTe	Deletes the specified audio analysis view.
DISPlay:AUDio:MEASview<y>:NEW	Displays a new audio analysis view.
DISPlay:AUDio:MEASview<y>:SELeCt	Selects an audio analysis view on the screen.
DISPlay:AUDio:SPECTrum:FREQUency:[SCALe]:STARt	Sets or queries the start frequency (left edge) of the audio spectrum graph.
DISPlay:AUDio:SPECTrum:FREQUency:[SCALe]:STOP	Sets or queries the stop frequency (right edge) of the audio spectrum graph.
DISPlay:AUDio:SPECTrum:FREQUency:AUTO	Rescales the horizontal axis automatically to fit the waveform to the screen in the audio spectrum view.
DISPlay:AUDio:SPECTrum:MARKer:SHOW:STATe	Determines whether to show or hide the readouts for the selected marker in the audio spectrum view.
DISPlay:AUDio:SPECTrum:RESet:SCALe	Resets the horizontal and vertical scales in the audio spectrum view.
DISPlay:AUDio:SPECTrum:SCALe:LOG:STATe	Determines whether or not to set the horizontal axis logarithmic in the audio spectrum view.
DISPlay:AUDio:SPECTrum:SHOW:NHARmonic:THReshold	Sets or queries the appearance of the non-harmonic threshold in the audio spectrum view.
DISPlay:AUDio:SPECTrum:TABLe:SHOW:STATe	Determines whether to show or hide the results table in the audio spectrum view.
DISPlay:AUDio:SPECTrum:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule grid on the screen.
DISPlay:AUDio:SPECTrum:Y:[SCALe]	Sets or queries the vertical scale in the audio spectrum view.
DISPlay:AUDio:SPECTrum:Y[:SCALe]:AUTO	Rescales the vertical axis automatically in the audio spectrum view.
DISPlay:AUDio:SPECTrum:Y[:SCALe]:OFFSet	Sets or queries the vertical position (offset) in the audio spectrum view.
<b>DISPlay:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
DISPlay:AVTime[:MEASview<y>]:LEGend:STATe	Shows or hides the trace legend in the amplitude versus time view.
DISPlay:AVTime[:MEASview<y>]:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:AVTime[:MEASview<y>]:RESet	Resets the horizontal and vertical scale to the default values.
DISPlay:AVTime[:MEASview<y>]:TRIGger:LEVel:STATe	Determines whether to show the power trigger level line on screen.
DISPlay:AVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:AVTime[:MEASview<y>]:X:RSCale	Rescales the horizontal axis automatically.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
Table continued...	



Command	Description
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:FULL	Sets or queries the horizontal scale.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:AVTime[:MEASview<y>]:Y:RSCale	Rescales the vertical axis automatically.
DISPlay:AVTime[:MEASview<y>]:Y[:SCALe]:FULL	Sets or queries the vertical scale.
DISPlay:AVTime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
DISPlay:CONSte[:MEASview<x>]:MPHase	Sets or queries the multiplication constant of the phase multiplication constellation display for a CPM signal. This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to CPM.
DISPlay:CONSte[:MEASview<x>]:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule grid on the screen.
<b>DISPlay:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
DISPlay:DDEMod[:MEASview<y>]:DELeTe	Deletes the measurement view in the general purpose digital modulation measurements.
DISPlay:DDEMod[:MEASview<y>][:SOURce<x>]:NEw	Displays a new measurement view.
DISPlay:DDEMod[:MEASview<y>]:SELeCt	Selects a measurement view in the general purpose digital modulation measurements on the screen.
DISPlay:DDEMod[:SOURce<x>]:RADix	Sets or queries the base of symbols of the measurement source.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:RESet	Presets the horizontal scale to the default value.
<b>DISPlay:DIAGram subgroup (Option 21 only)</b>	<b>Eye/Trellis diagram</b>
DISPlay:DIAGram:X[:SCALe]	Sets or queries the horizontal range.
DISPlay:DIAGram:X[:SCALe]:RESet	Presets the horizontal scale to the default value.
<b>DISPlay:DIQVtime subgroup (Option 21 only)</b>	<b>Demodulated I&amp;Q versus Time measurement</b>

Table continued...

Command	Description
DISPlay:DIQVtime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid on the screen.
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis).
<b>DISPlay:DPX subgroup</b>	<b>DPXogram measurement</b>
DISPlay:DPX:DGRam:TIME[:SCALe]:OFFSet:DIVision	Sets or queries the DPXogram vertical time offset in divisions.
DISPlay:DPX:DGRam:TIME[:SCALe]:PDIVision	Sets or queries the DPXogram vertical time scale per division.
DISPlay:DPX:DGRam:TIME[:SCALe]:RESet	Presets the time scale to the default value for the DPXogram measurement.
DISPlay:DPX:DGRam:TSTamp:STATe	Determines whether to display the time stamp readout in the DPXogram display.
DISPlay:DPX:DGRam:Y[:SCALe]:AUTO	Rescales the height axis automatically to fit the waveform to the screen in the DPXogram display.
DISPlay:DPX:DGRam:Y[:SCALe]:RESet	Resets the height scale of the DPXogram display.
DISPlay:DPX:LEGenD:STATe	Determines whether to show or hide the trace legend on the display.
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS	Sets or queries the vertical axis representation.
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS:OFFSet	Sets or queries the vertical offset in the DPX Phase view.
DISPlay:DPX:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:DPX:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division).
DISPlay:DPX:WINDow:TRACe:GRATICule:GRID:STATe	Shows or hides the graticule grid on the screen
<b>DISPlay:EDIagram subgroup (Option 21 only)</b>	<b>Eye diagram</b>
DISPlay:EDIagram[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule grid on the screen in the eye diagram.
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical range of the eye diagram.
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen in the eye diagram.
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis) of the eye diagram.
<b>DISPlay:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
DISPlay:EVM[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule grid view state for the EVM view.
DISPlay:EVM[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical range of the EVM versus Time graph.
DISPlay:EVM[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen in the EVM versus Time display.
DISPlay:EVM[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge) of the EVM versus Time graph.
<b>DISPlay:FDVTime subgroup (Option 21 only)</b>	<b>Frequency deviation versus Time measurement</b>
DISPlay:FDVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule grid view state for the FDV view.

Table continued...

Command	Description
DISPlay:FDVTime[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical range of the Frequency deviation versus Time graph.
DISPlay:FDVTime[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen in the Frequency deviation versus Time display.
DISPlay:FDVTime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge) of the Frequency deviation versus Time graph.
<b>DISPlay:{FSETtling PSETtling} subgroup</b>	<b>Frequency and Phase Settling measurements</b>
DISPlay:{FSETtling PSETtling}:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker in the Frequency or Phase Settling views.
DISPlay:{FSETtling PSETtling}:TIME:DECimal	Sets or queries the number of values to the right of the decimal point.
DISPlay:{FSETtling PSETtling}:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule grid on the screen.
DISPlay:{FSETtling PSETtling}:X[:SCALe]	Sets or queries the horizontal scale (full-scale time) of the Settling Time graph.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:AUTO	Sets the horizontal scale automatically to fit the waveform to the screen in the Frequency and Phase Settling views.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge) of the Frequency versus Time graph.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:{FSETtling PSETtling}:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:{FSETtling PSETtling}:Y[:SCALe]	Sets or queries the vertical range of the Frequency versus Time graph.
DISPlay:{FSETtling PSETtling}:Y[:SCALe]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen in the Frequency and Phase Settling views.
DISPlay:{FSETtling PSETtling}:Y[:SCALe]:OFFSet	Sets or queries the vertical offset (the value at the center of the vertical axis) in the Frequency and Phase Settling graphs.
DISPlay:{FSETtling PSETtling}:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division) of the Frequency and Phase Settling graphs.
<b>DISPlay:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
DISPlay:FVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid on the Frequency versus Time view.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.

Table continued...

Command	Description
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:FVTime[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:FVTime[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:FVTime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:GENeral subgroup</b>	<b>General signal viewing</b>
DISPlay:GENeral:MEASview<y>:DELete	Deletes the measurement view.
DISPlay:GENeral:MEASview<y>[:SOURce<x>]:NEW	Displays a new measurement view.
DISPlay:GENeral:MEASview<y>:SELect	Sets or queries the measurement view.
<b>DISPlay:GPRF subgroup</b>	<b>General purpose RF measurements</b>
DISPlay:GPRF:MEASview:DELete	Deletes the measurement view.
DISPlay:GPRF:MEASview[:SOURce<x>]:NEW	Displays a new measurement view.
DISPlay:GPRF:MEASview:SELect	Sets or queries the measurement view.
<b>DISPlay:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
DISPlay:IQVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid on the screen.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:IQVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALe]:RESCale	Rescales the vertical scale.
<b>DISPlay:MCPower subgroup</b>	<b>MCPR measurement</b>
DISPlay:MCPower:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:MCPower:PLEVel:SHOW:STATe	Determines whether to show the power levels.
DISPlay:MCPower:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:MCPower:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:MCPower:X[:SCALe]	Sets or queries the horizontal range.

Table continued...

Command	Description
DISPlay:MCPower:X[:SCALE]:AUTO	Rescales the horizontal axis automatically.
DISPlay:MCPower:X[:SCALE]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:MCPower:Y[:SCALE]	Sets or queries the vertical range.
DISPlay:MCPower:Y[:SCALE]:AUTO	Rescales the vertical axis automatically.
DISPlay:MCPower:Y[:SCALE]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
DISPlay:MERRor[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule grid view state for the Magnitude error versus Time view.
DISPlay:MERRor[:MEASview<y>]:Y[:SCALE]	Sets or queries the vertical range of the Magnitude error versus Time graph.
DISPlay:MERRor[:MEASview<y>]:Y[:SCALE]:AUTO	Sets the vertical scale automatically to fit the waveform to the screen in the Magnitude error versus Time display.
DISPlay:MERRor[:MEASview<y>]:Y[:SCALE]:OFFSet	Sets or queries the minimum vertical value (bottom edge) of the Magnitude error versus Time graph.
DISPlay:OBWidth subgroup	<b>Occupied Bandwidth measurement</b>
DISPlay:OBWidth:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:OBWidth:RESet:SCALE	Resets the horizontal and vertical scale to the default values.
DISPlay:OBWidth:SELected:BANDwidth	Sets or queries the bandwidth (OBW or x dB BW) to measure.
DISPlay:OBWidth:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:OBWidth:X[:SCALE]	Sets or queries the horizontal range.
DISPlay:OBWidth:X[:SCALE]:AUTO	Rescales the horizontal axis automatically.
DISPlay:OBWidth:X[:SCALE]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:OBWidth:Y[:SCALE]	Sets or queries the vertical range
DISPlay:OBWidth:Y[:SCALE]:AUTO	Rescales the vertical axis automatically.
DISPlay:OBWidth:Y[:SCALE]:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM subgroup	<b>OFDM measurements</b>
DISPlay:OFDM:CONSte[:SCALE]	Sets or queries the horizontal offset.
DISPlay:OFDM:CONSte:X:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM:CONSte:Y:OFFSet	Sets or queries the scale.
DISPlay:OFDM:CRESpone:FREQuency:AUTO	Rescales the horizontal axis automatically.
DISPlay:OFDM:CRESpone:FREQuency:OFFSet	Sets or queries the frequency offset.
DISPlay:OFDM:CRESpone:MAGNitude:AUTO	Sets or queries the horizontal range.
DISPlay:OFDM:CRESpone:MAGNitude:OFFSet	Rescales the magnitude automatically.
DISPlay:OFDM:CRESpone:MAGNitude:PDIVision	Sets or queries the magnitude offset.
DISPlay:OFDM:CRESpone:PHASe:AUTO	Sets or queries the scale (per division) of the magnitude.
DISPlay:OFDM:CRESpone:PHASe:OFFSet	Rescales the vertical axis automatically.
DISPlay:OFDM:CRESpone:PHASe[:SCALE]	Sets or queries the phase offset.
DISPlay:OFDM:CRESpone:WINDow:SELEct:PLOT	Sets the vertical scale automatically.

Table continued...

Command	Description
DISPlay:OFDM:CRESPonse:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the plot
DISPlay:OFDM:EVM:FREQUency[:SCALE]	Determines whether to show or hide the graticule.
DISPlay:OFDM:EVM:FREQUency:AUTO	Sets or queries the horizontal range.
DISPlay:OFDM:EVM:FREQUency:OFFSet	Rescales the frequency automatically to fit the waveform to the screen.
DISPlay:OFDM:EVM:MARKer:SHOW:STATe	Sets or queries the frequency offset.
DISPlay:OFDM:EVM:TIME[:SCALE]	Determines whether to show or hide the readout.
DISPlay:OFDM:EVM:TIME:AUTO	Rescales the time automatically to fit the waveform to the screen..
DISPlay:OFDM:EVM:TIME:OFFSet	Sets or queries the time offset.
DISPlay:OFDM:EVM:WINDow:SELect:PLOT	Sets or queries the plot.
DISPlay:OFDM:EVM:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:OFDM:EVM:Y[:SCALE]	Sets or queries the vertical range.
DISPlay:OFDM:EVM:Y:AUTO	Rescales the vertical scale automatically.
DISPlay:OFDM:EVM:Y:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM:FLATness:AUTO	Automatically rescales the horizontal and vertical axes for the best display.
DISPlay:OFDM:FLATness:X:AUTO	Rescales the horizontal axis automatically in the OFDM Spectral Flatness display.
DISPlay:OFDM:FLATness:X:OFFSet	Sets or queries the value of the offset of the horizontal axis.
DISPlay:OFDM:FLATness:X[:SCALE]	Sets or queries the value of the horizontal scale.
DISPlay:OFDM:FLATness:Y:AUTO	Rescales the vertical axis scale and position values automatically.
DISPlay:OFDM:FLATness:Y:OFFSet	Sets or queries the value of the offset from the center (vertical position).
DISPlay:OFDM:FLATness:Y[:SCALE]	Sets or queries the vertical scale value.
DISPlay:OFDM:MEASview:DELete	Deletes the specified OFDM measurement view.
DISPlay:OFDM:MEASview:NEW	Creates a new OFDM measurement view.
DISPlay:OFDM:MEASview:SELect	Sets or queries the OFDM measurement view.
DISPlay:OFDM:MERRor:FREQUency[:SCALE]	Sets or queries the horizontal range.
DISPlay:OFDM:MERRor:FREQUency[:SCALE]:AUTO	Rescales the frequency automatically to fit the waveform to the screen.
DISPlay:OFDM:MERRor:FREQUency[:SCALE]:OFFSet	Sets or queries the frequency offset.
DISPlay:OFDM:MERRor:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:OFDM:MERRor:TIME[:SCALE]	Sets or queries the time scale.
DISPlay:OFDM:MERRor:TIME[:SCALE]:AUTO	Rescales the time automatically to fit the waveform to the screen.
DISPlay:OFDM:MERRor:TIME[:SCALE]:OFFSet	Sets or queries the time offset.
DISPlay:OFDM:MERRor:WINDow:SELect:PLOT	Sets or queries the plot.
DISPlay:OFDM:MERRor:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.

Table continued...

Command	Description
DISPlay:OFDM:MERRor:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:OFDM:MERRor:Y:AUTO	Rescales the vertical scale automatically to fit the waveform.
DISPlay:OFDM:MERRor:Y:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM:PERRor:FREQUency[:SCALe]	Sets or queries the horizontal range.
DISPlay:OFDM:PERRor:FREQUency[:SCALe]:AUTO	Rescales the frequency automatically to fit the waveform to the screen.
DISPlay:OFDM:PERRor:FREQUency[:SCALe]:OFFSet	Sets or queries the frequency offset.
DISPlay:OFDM:PERRor:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:OFDM:PERRor:TIME[:SCALe]	Sets or queries the time scale.
DISPlay:OFDM:PERRor:TIME[:SCALe]:AUTO	Rescales the time automatically to fit the waveform to the screen.
DISPlay:OFDM:PERRor:TIME[:SCALe]:OFFSet	Sets or queries the time offset.
DISPlay:OFDM:PERRor:WINDow:SElect:PLOT	Sets or queries the plot.
DISPlay:OFDM:PERRor:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:OFDM:PERRor:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:OFDM:PERRor:Y:AUTO	Rescales the vertical scale automatically to fit the waveform.
DISPlay:OFDM:PERRor:Y:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM:POWER:FREQUency[:AUTO]	Rescales the frequency automatically to fit the waveform to the screen.
DISPlay:OFDM:POWER:FREQUency[:OFFSet]	Sets or queries the frequency offset.
DISPlay:OFDM:POWER:FREQUency[:SCALe]	Sets or queries the horizontal range.
DISPlay:OFDM:POWER:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:OFDM:POWER:TIME[:AUTO]	Rescales the time automatically to fit the waveform to the screen.
DISPlay:OFDM:POWER:TIME:OFFSet	Sets or queries the time offset.
DISPlay:OFDM:POWER:TIME[:SCALe]	Sets or queries the time scale.
DISPlay:OFDM:POWER:WINDow:SElect:PLOT	Sets or queries the plot.
DISPlay:OFDM:POWER:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:OFDM:POWER:Y:AUTO	Rescales the vertical scale automatically to fit the waveform.
DISPlay:OFDM:POWER:Y:OFFSet	Sets or queries the vertical offset.
DISPlay:OFDM:POWER:Y[:SCALe]	Sets or queries the vertical range.
<b>DISPlay:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
DISPlay:PERRor[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries the graticule grid view state for the Phase error versus Time view.
DISPlay:PERRor[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:PERRor[:MEASview<y>]Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:PERRor[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the minimum vertical value (bottom edge).
<b>DISPlay:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>

Table continued...

Command	Description
DISPlay:PHVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid on the screen.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:AUTO	Sets the horizontal scale automatically.
DISPlay:PHVTime[:MEASview?[:Y[:SCALe]:AXIS:ENABle:REFereNce	Sets or queries the state of <b>Y Reference</b> checkbox.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe	Determines whether to set the horizontal scale automatically or manually.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MAXimum?	Queries the upper limit of the horizontal scale setting range.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MINimum?	Queries the lower limit of the horizontal scale setting range.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum?	Queries the upper limit of the horizontal offset setting range.
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?	Queries the lower limit of the horizontal offset setting range.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale automatically.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]:AXIS	Sets or queries the vertical axis representation.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]:AXIS:REFereNce	Sets or queries the reference time for phase.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALe]:RESCale	Rescales the vertical scale.
<b>DISPlay:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
DISPlay:PNOise:LEGenD:STATe	Sets or queries showing the trace legend on the display.
DISPlay:PNOise:MARKer:SHOW:STATe	Determines whether to show the readout for the selected marker.
DISPlay:PNOise:RESet:SCALe	Resets the horizontal and vertical scale to the default values.
DISPlay:PNOise:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid on the screen.
DISPlay:PNOise:X[:SCALe]:AUTO	Rescales the horizontal axis automatically.
DISPlay:PNOise:X[:SCALe]:START	Sets or queries the start frequency of the graph.
DISPlay:PNOise:X[:SCALe]:STOP	Sets or queries the stop frequency of the graph.
DISPlay:PNOise:Y[:SCALe]	Sets or queries the stop frequency of the graph.
DISPlay:PNOise:Y[:SCALe]:AUTO	Rescales the vertical axis automatically.
DISPlay:PNOise:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:PNOise:Y[:SCALe]:PDIVision	Sets or queries the vertical scale (per division).
<b>DISPlay:PULSe subgroup</b>	<b>Pulsed RF measurements</b>
DISPlay:PULSe:MEASview<y>:DELete	Deletes the measurement view.
DISPlay:PULSe:MEASview<y>[:SOURce<x>]:NEW	Displays a new measurement view.
DISPlay:PULSe:MEASview<y>:SELect	Selects the specified Measurement view.
DISPlay:PULSe:RESult[:MEASview<y>]:ATX	Sets or queries whether to show the average transmitted power result.
DISPlay:PULSe:RESult[:MEASview<y>]:AVERAge	Sets or queries whether to show the average on power result.

Table continued...



Command	Description
DISPlay:PULSe:RESult[:MEASview<y>]:AVERage	Sets or queries whether to show the average on power result.
DISPlay:PULSe:RESult[:MEASview<y>]:DRODb	Sets or queries whether to show the droop in the results table.
DISPlay:PULSe:RESult[:MEASview<y>]:DROop	Sets or queries showing the droop measurement result in the pulse table
DISPlay:PULSe:RESult[:MEASview<y>]:DUTPct	Sets or queries whether to show the duty factor (%) result.
DISPlay:PULSe:RESult[:MEASview<y>]:DUTRatio	Sets or queries whether to show the duty factor (ratio) result.
DISPlay:PULSe:RESult[:MEASview<y>]:FALL	Sets or queries whether to show the fall time in the results table.
DISPlay:PULSe:RESult[:MEASview<y>]:FDELta	Sets or queries whether to show the delta frequency result.
DISPlay:PULSe:RESult[:MEASview<y>]:FRDeviatiOn	Sets or queries whether to show the frequency deviation result.
DISPlay:PULSe:RESult[:MEASview<y>]:IRAmplitude	Sets or queries showing the Impulse Response Amplitude measurement result in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:IRTime	Sets or queries showing the Impulse Response Time measurement result in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:MFRreqerror	Sets or queries whether to show the maximum frequency error result.
DISPlay:PULSe:RESult[:MEASview<y>]:MPHerror	Sets or queries whether to show the maximum phase error result.
DISPlay:PULSe:RESult[:MEASview<y>]:OVEDb	Sets or queries whether to show the Overshoot measurement result in dB in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:OVERshoot	Sets or queries whether to show the Overshoot measurement result in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:PHDeviatiOn	Sets or queries whether to show the phase deviation result.
DISPlay:PULSe:RESult[:MEASview<y>]:PPFRequency	Sets or queries whether to show the pulse-pulse frequency result.
DISPlay:PULSe:RESult[:MEASview<y>]:PPOWer	Sets or queries whether to show the peak power in the results table.
DISPlay:PULSe:RESult[:MEASview<y>]:PPPHase	Sets or queries whether to show the pulse-pulse carrier phase result.
DISPlay:PULSe:RESult[:MEASview<y>]:RINterval	Sets or queries whether to show the repetition interval result.
DISPlay:PULSe:RESult[:MEASview<y>]:RIPDb	Sets or queries showing the ripple measurement result in dB in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:RIPPlE	Sets or queries showing the ripple measurement result in dB in the pulse table.
DISPlay:PULSe:RESult[:MEASview<y>]:RISE	Sets or queries whether to show the rise time in the results table.
DISPlay:PULSe:RESult[:MEASview<y>]:RMSFreqerror	Sets or queries whether to show the RMS frequency error result.
DISPlay:PULSe:RESult[:MEASview<y>]:RMSPherror	Sets or queries whether to show the RMS phase error result.
DISPlay:PULSe:RESult[:MEASview<y>]:RRATE	Sets or queries whether to show the repetition rate result.
DISPlay:PULSe:RESult[:MEASview<y>]:TIME	Sets or queries whether to show the time in the results table.
DISPlay:PULSe:RESult[:MEASview<y>]:WIDTh	Sets or queries whether to show the pulse width in the results table.
DISPlay:PULSe:SElect[:MEASview<y>]:NUMBer	Sets or queries the pulse to measure.
DISPlay:PULSe:SElect[:MEASview<y>]:RESult	Sets or queries which result is shown in the trace and statistics views.

Table continued...

Command	Description
DISPlay:PULSe:STATistics[:MEASview<y>]:MARKer<x>:SHOW:STATe	Sets or queries whether to show the marker readout in the graph.
DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT	Sets or queries how to show the statistics graph.
DISPlay:PULSe:STATistics[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid in the statistics view.
DISPlay:PULSe:STATistics[:MEASview<y>]:X:RSCale	Rescales the horizontal axis of the statistics graph.
DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALe]:NUMBer	Sets or queries the horizontal scale (the number of pulses per division).
DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value in the statistics view.
DISPlay:PULSe:STATistics[:MEASview<y>]:Y:RSCale	Rescales the vertical axis of the statistics graph.
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALe]:FULL	Sets or queries the vertical full-scale in the statistics view.
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset in the statistics view.
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALe]:STOP?	Queries the minimum vertical value in the statistics view.
DISPlay:PULSe:TRACe[:MEASview<y>]:MARKer<x>:SHOW:STATe	Sets or queries whether to show the marker readout in the trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:POINT:SHOW	Sets or queries whether to show the measurement points and lines.
DISPlay:PULSe:TRACe[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show the graticule grid in the trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:X:RSCale	Rescales the horizontal axis of the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]	Sets or queries the horizontal full scale in the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:FULL	Sets or queries the full-scale reference for the horizontal rescale.
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value in the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:PDIVision	Sets or queries the horizontal full scale in the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:Y:RSCale	Rescales the vertical axis of the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:FULL	Sets or queries the vertical full scale in the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset in the pulse trace view.
DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:STOP?	Queries the minimum vertical value in the pulse trace view.
<b>DISPlay:SEM subgroup</b>	<b>Spectral Emissions Mask measurements</b>
DISPlay:SEM:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:SEM:RESet:SCALe	Resets the scale.
DISPlay:SEM:SHOW:LIMit	Sets or queries the appearance of the limits.
DISPlay:SEM:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:SEM:X[:SCALe]:AUTO	Rescales the horizontal scale automatically to fit the waveform.
DISPlay:SEM:X[:SCALe]:START	Sets or queries the start frequency.
DISPlay:SEM:X[:SCALe]:STOP	Sets or queries the stop frequency.
DISPlay:SEM:Y[:SCALe]	Sets or queries the vertical range.
DISPlay:SEM:Y[:SCALe]:AUTO	Rescales the vertical scale automatically to fit the waveform.

Table continued...

Command	Description
DISPlay:SEM:Y[:SCALE]:OFFSet	Sets or queries the vertical offset.
<b>DISPlay:SGRam subgroup</b>	<b>Spectrogram measurement</b>
DISPlay:SGRam[:MEASview<y>]:FREQUency:AUTO	Rescales the horizontal (frequency) axis automatically.
DISPlay:SGRam[:MEASview<y>]:FREQUency:OFFSet	Sets or queries the horizontal (frequency) offset.
DISPlay:SGRam[:MEASview<y>]:FREQUency:SCALE	Sets or queries the horizontal (frequency) range.
DISPlay:SGRAM[:MEASview<y>]:MARKer:SHOW:STATe	Sets or queries the marker readout.
DISPlay:SGRAM[:MEASview<y>]:SELected:TIMeStamp	Sets or queries the time stamp readout.
DISPlay:SGRam[:MEASview<y>]:TIME:AUTO	Rescales the vertical axis automatically.
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet	Sets or queries the vertical axis (time) offset (bottom line number).
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet:DIVisions	Sets or queries the Vertical offset or Position in divisions.
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent	Sets or queries the Overlap in percent.
DISPlay:SGRam[:MEASview<y>]:TIME:SCALE	Sets or queries the vertical scale (the amount of time in each line).
DISPlay:SGRam[:MEASview<y>]:TIME:SCALE:PER:DIVision	Sets or queries the vertical Time/division value.
DISPlay:SGRam[:MEASview<y>]:TIME:SPECTrums:PERLine?	Queries the Spectrums/Line value.
<b>DISPlay:SPECTrum subgroup</b>	<b>Spurious measurement</b>
DISPlay:SPURious:MARKer:SHOW:STATe	Sets or queries whether to show the readout for the selected marker.
DISPlay:SPURious:RESet:SCALE	Resets the horizontal and vertical scale to the default values.
DISPlay:SPURious:SCALE:LOG:STATe	Sets or queries whether to set the horizontal axis logarithmic.
DISPlay:SPURious:SELect:NUMBER	Sets or queries the spurious number.
DISPlay:SPURious:SHOW:LIMit	Sets or queries how to display the limits.
DISPlay:SPURious:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show the graticule grid on screen.
DISPlay:SPURious:X[:SCALE]:AUTO	Rescales the horizontal axis automatically.
DISPlay:SPURious:X[:SCALE]:STARt	Sets or queries the minimum horizontal value of the spectrum graph.
DISPlay:SPURious:X[:SCALE]:STOP	Sets or queries the maximum horizontal value of the spectrum graph.
DISPlay:SPURious:Y[:SCALE]	Sets or queries the vertical range of the spectrum graph.
DISPlay:SPURious:Y[:SCALE]:AUTO	Rescales the vertical axis automatically.
DISPlay:SPURious:Y[:SCALE]:OFFSet	Sets or queries the vertical offset of the spectrum graph.
<b>DISPlay:TDiagram subgroup (Option 21 only)</b>	<b>Trellis diagram</b>
DISPlay:TDiagram[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe	Sets or queries whether to show or hide the graticule grid on the screen in the Trellis diagram.
DISPlay:TDiagram[:MEASview<y>]:Y[:SCALE]	Sets or queries the vertical scale of the Trellis diagram.
DISPlay:TDiagram[:MEASview<y>]:Y[:SCALE]:AUTO	Sets the vertical scale automatically.
DISPlay:TDiagram[:MEASview<y>]:Y[:SCALE]:OFFSet	Sets or queries the vertical offset (center point of the vertical axis).
<b>DISPlay:TOVerview subgroup</b>	<b>Time overview</b>
Table continued...	

Command	Description
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe	Sets or queries whether to show the graticule grid in the time overview.
DISPlay:TOVerview[:MEASview<y>]:X[:SCALe]	Sets or queries the horizontal scale.
DISPlay:TOVerview[:MEASview<y>]:X[:SCALe]:AUTO	Sets the horizontal scale and offset automatically.
DISPlay:TOVerview[:MEASview<y>]:X[:SCALe]:OFFSet	Sets or queries the minimum horizontal value (left edge).
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]	Sets or queries the vertical scale.
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]:AUTO	Sets the vertical scale and offset automatically.
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]:OFFSet	Sets or queries the vertical offset.
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]:RESCale	Rescales the vertical scale.
<b>DISPlay:WINDow subgroup</b>	<b>Window display</b>
DISPlay:WINDow:ACQuisition:BANdwidth:SHOw:STATe	Sets or queries the state of <b>Show Acq Bandwidth</b> checkbox.
DISPlay:WINDow:ACTive:MEASurement?	Queries the active measurement views
DISPlay:WINDow:APPLY:SOURce:COLOr	Sets or queries the state of <b>Apply Source Color to Trace</b> checkbox.
DISPlay:WINDow:COLor:SCHeme	Sets or queries the color scheme for displaying traces and background on the screen.
DISPlay:WINDow:OPTimized:MEASurement?	Queries the measurement views that are optimized.
<b>DISPlay:WLAN subgroup</b>	<b>Wireless LAN measurements</b>
DISPlay:WLAN:CONSte[:SCALe]	Sets the magnification value (zoom factor).
DISPlay:WLAN:CONSte:X:OFFSet	Sets or queries the value of the offset of the horizontal axis.
DISPlay:WLAN:CONSte:Y:OFFSet	Sets or queries the value of the offset of the vertical axis.
DISPlay:WLAN:CRESPonse:AUTO	Automatically rescales the horizontal and vertical values.
DISPlay:WLAN:CRESPonse:FREQuency:AUTO	Automatically rescales the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:CRESPonse:FREQuency:START	Specifies or queries the horizontal start value.
DISPlay:WLAN:CRESPonse:FREQuency:STOP	Specifies or queries the horizontal stop value.
DISPlay:WLAN:CRESPonse:MAGNitude:AUTO	Rescales the vertical magnitude offset (position) and scale values.
DISPlay:WLAN:CRESPonse:MAGNitude:OFFSet	Sets or queries the vertical offset.
DISPlay:WLAN:CRESPonse:MAGNitude[:SCALe]	Sets or queries the value of the vertical scale.
DISPlay:WLAN:CRESPonse:MARKer:SHOW:STATe	Shows or hides the marker readout.
DISPlay:WLAN:CRESPonse:PHASe:AUTO	Rescales the vertical phase value automatically to fit the waveform to the screen.
DISPlay:WLAN:CRESPonse:PHASe:OFFSet	Sets or queries the vertical offset.
DISPlay:WLAN:CRESPonse:PHASe[:SCALe]	Sets or queries the phase value.
DISPlay:WLAN:CRESPonse:WINDow:SELEct:PLOT	Specifies or queries which graphs are displayed.
DISPlay:WLAN:CRESPonse:WINDow:TRACe:GRATicule:GRID:STATe	Sets or queries whether to show the graticule grid.
DISPlay:WLAN:EVM:FREQuency:AUTO	Automatically rescales the horizontal subcarrier (frequency) start and stop values.

Table continued...

Command	Description
DISPlay:WLAN:EVM:FREQuency:RESet	Automatically resets the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:EVM:FREQuency:STARt	Specifies or queries the horizontal start value.
DISPlay:WLAN:EVM:FREQuency:STOP	Specifies or queries the horizontal stop value.
DISPlay:WLAN:EVM:MARKer:SHOW:STATe	Shows or hides the Marker readout.
DISPlay:WLAN:EVM[:SCALE]:AUTO	Automatically adjusts all vertical and horizontal values.
DISPlay:WLAN:EVM:TIME:AUTO	Sets the horizontal start and stop values to fit the waveform.
DISPlay:WLAN:EVM:TIME:AUTO:STATe	Automatically sets the horizontal symbol scale value.
DISPlay:WLAN:EVM:TIME:RESet	Resets the horizontal symbol start and stop values.
DISPlay:WLAN:EVM:TIME:STARt	Sets or queries the horizontal symbol start value.
DISPlay:WLAN:EVM:TIME:STOP	Sets or queries the horizontal symbol stop value
DISPlay:WLAN:EVM:WINDow:SELEct:PLOT	Specifies or queries which graphs are displayed in the WLAN EVM display.
DISPlay:WLAN:EVM:WINDow:TRACe:GRATICule:GRID:STATe	Shows or hides the graticule grid on the screen.
DISPlay:WLAN:EVM:Y:AUTO	Automatically sets the vertical scale and position values.
DISPlay:WLAN:EVM:Y:OFFSet	Sets or queries the value of the offset from the center.
DISPlay:WLAN:EVM:Y:RESet	Resets the vertical scale and position (offset) values.
DISPlay:WLAN:EVM:Y[:SCALE]	Sets or queries the vertical scale value.
DISPlay:WLAN:FLATness:AUTO	Automatically rescales the horizontal and vertical axes.
DISPlay:WLAN:FLATness:X:AUTO	Automatically rescales the horizontal axis.
DISPlay:WLAN:FLATness:X:OFFSet	Sets or queries the value of the offset of the horizontal axis from the center.
DISPlay:WLAN:FLATness:X[:SCALE]	Sets or queries the horizontal scale value.
DISPlay:WLAN:FLATness:Y:AUTO	Automatically rescales the vertical axis scale and position values.
DISPlay:WLAN:FLATness:Y:OFFSet	Sets or queries the value of the offset from the center.
DISPlay:WLAN:FLATness:Y[:SCALE]	Sets or queries the vertical scale value.
DISPlay:WLAN:MEASview:DELEte	Deletes the selected measurement view.
DISPlay:WLAN:MEASview:NEW	Displays a new measurement view.
DISPlay:WLAN:MEASview:SELEct	Selects a WLAN measurement view.
DISPlay:WLAN:MERRor:FREQuency:AUTO	Automatically rescales the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:MERRor:FREQuency:RESet	Automatically resets the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:MERRor:FREQuency:STARt	Specifies or queries the horizontal start value.
DISPlay:WLAN:MERRor:FREQuency:STOP	Specifies or queries the horizontal stop value.
DISPlay:WLAN:MERRor:MARKer:SHOW:STATe	Shows or hides the marker readout.
DISPlay:WLAN:MERRor[:SCALE]:AUTO	Automatically adjusts all vertical and horizontal values.
DISPlay:WLAN:MERRor:TIME:AUTO	Sets the horizontal start and stop values.

Table continued...

Command	Description
DISPlay:WLAN:MERRor:TIME:AUTO:STATe	Automatically sets the horizontal symbol scale value.
DISPlay:WLAN:MERRor:TIME:RESet	Resets the horizontal symbol start and stop values.
DISPlay:WLAN:MERRor:TIME:START	Sets or queries the horizontal symbol start value.
DISPlay:WLAN:MERRor:TIME:STOP	Sets or queries the horizontal symbol stop value.
DISPlay:WLAN:MERRor:WINDow:SElect:PLOT	Specifies or queries which graphs are displayed.
DISPlay:WLAN:MERRor:WINDow:TRACe:GRATICule:GRID:STATe	Shows or hides the graticule grid on the display.
DISPlay:WLAN:MERRor:Y:AUTO	Automatically sets the vertical scale and position values.
DISPlay:WLAN:MERRor:Y:OFFSet	Sets or queries the offset from the center (vertical position).
DISPlay:WLAN:MERRor:Y:RESet	Resets the vertical scale and position (offset).
DISPlay:WLAN:MERRor:Y[:SCALE]	Sets or queries the vertical scale value.
DISPlay:WLAN:PERRor:FREQuency:AUTO	Automatically rescales the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:PERRor:FREQuency:RESet	Automatically resets the horizontal subcarrier (frequency) start and stop values.
DISPlay:WLAN:PERRor:FREQuency:START	Specifies or queries the horizontal start value.
DISPlay:WLAN:PERRor:FREQuency:STOP	Specifies or queries the horizontal stop value.
DISPlay:WLAN:PERRor:MARKer:SHOW:STATe	Show or hides the Marker readout.
DISPlay:WLAN:PERRor[:SCALE]:AUTO	Automatically adjusts all vertical and horizontal settings.
DISPlay:WLAN:PERRor:TIME:AUTO	Automatically sets the horizontal symbol start and stop values.
DISPlay:WLAN:PERRor:TIME:AUTO:STATe	Automatically sets the horizontal symbol scale value.
DISPlay:WLAN:PERRor:TIME:RESet	Resets the horizontal symbol start and stop values.
DISPlay:WLAN:PERRor:TIME:START	Sets or queries the horizontal symbol start value.
DISPlay:WLAN:PERRor:TIME:STOP	Sets or queries the horizontal symbol stop value.
DISPlay:WLAN:PERRor:WINDow:SElect:PLOT	Specifies or queries which displays are displayed.
DISPlay:WLAN:PERRor:WINDow:TRACe:GRATICule:GRID:STATe	Shows or hides the graticule grid on the screen.
DISPlay:WLAN:PERRor:Y:AUTO	Automatically selects the vertical scale and position values.
DISPlay:WLAN:PERRor:Y:OFFSet	Sets or queries the value of the offset from the center (vertical position).
DISPlay:WLAN:PERRor:Y:RESet	Resets the vertical scale and position (offset) values.
DISPlay:WLAN:PERRor:Y[:SCALE]	Sets or queries the vertical scale value.
DISPlay:WLAN:PVTIme:BURSt:X[:SCALE]	Sets or queries the value of the scale (width).
DISPlay:WLAN:PVTIme:BURSt:X[:SCALE]:AUTO	Automatically sets the starting time (position) and scale (width) values
DISPlay:WLAN:PVTIme:BURSt:X[:SCALE]:OFFSet	Sets or queries the starting time (position) value.
DISPlay:WLAN:PVTIme:BURSt:X[:SCALE]:RESet	Resets the starting time (position) and scale (width) values.
DISPlay:WLAN:PVTIme:FALL:X[:SCALE]	Sets or queries the scale (width) value.
DISPlay:WLAN:PVTIme:FALL:X[:SCALE]:AUTO	Automatically sets the starting time (position) and scale (width) values.

Table continued...

Command	Description
DISPlay:WLAN:PVTIme:FALL:X[:SCALe]:OFFSet	Sets or queries the value for the starting time (offset).
DISPlay:WLAN:PVTIme:FALL:X[:SCALe]:RESet	Resets the starting time (position) and scale (width) values.
DISPlay:WLAN:PVTIme:MARKer:SHOW:STATe	Shows or hides the marker readout.
DISPlay:WLAN:PVTIme:RISE:X[:SCALe]	Sets or queries the scale (width) value.
DISPlay:WLAN:PVTIme:RISE:X[:SCALe]:AUTO	Automatically sets the starting time (position) and scale (width) values.
DISPlay:WLAN:PVTIme:RISE:X[:SCALe]:OFFSet	Sets or queries the starting time value (offset).
DISPlay:WLAN:PVTIme:RISE:X[:SCALe]:RESet	Resets the starting time (position) and scale (width) values.
DISPlay:WLAN:PVTIme:WINDow:SELEct:PLOT	Specifies or queries which view to use.
DISPlay:WLAN:PVTIme:WINDow:TRACe:GRATICule:GRID:STATe	Shows or hides the graticule grid on the screen.
DISPlay:WLAN:PVTIme:Y[:SCALe]	Sets or queries the vertical scale value.
DISPlay:WLAN:PVTIme:Y[:SCALe]:AUTO	Automatically selects the vertical scale and position values.
DISPlay:WLAN:PVTIme:Y[:SCALe]:OFFSet	Sets or queries the value of the vertical offset.
DISPlay:WLAN:PVTIme:Y[:SCALe]:PDIVision	Sets or queries the value of the vertical scale.
DISPlay:WLAN:PVTIme:Y[:SCALe]:RESet	Resets the vertical scale and position (offset) values.
DISPlay:WLAN:SUMMary:EVMUnits	Sets or queries the EVM Units setting.
<b>DISPlay:LTE subgroup</b>	
DISPlay:LTE:ACLR:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:LTE:ACLR:RESet:SCALe	Resets the horizontal and vertical scale to default values.
DISPlay:LTE:ACLR:SHOW:LIMit	Sets or queries the appearance of the limits.
DISPlay:LTE:ACLR:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticules.
DISPlay:LTE:ACLR:WINDow:TRACe:POWEr:LEVEl:STATe	Determines whether to show or hide the power readout.
DISPlay:LTE:ACLR:X:SCALe	Sets or queries the horizontal range.
DISPlay:LTE:ACLR:X:SCALe:AUTO	Rescales the horizontal axis automatically to fit the waveform to the screen.
DISPlay:LTE:ACLR:X:SCALe:OFFSet	Sets or queries the center frequency of the trace.
DISPlay:LTE:ACLR:Y:SCALe:AUTO	Rescales the vertical scale automatically to fit the waveform on the screen.
DISPlay:LTE:ACLR:Y:SCALe:OFFSet	Sets or queries the vertical offset of the trace.
DISPlay:LTE:ACLR:Y:SCALe	Sets or queries the vertical range of the graph.
DISPlay:LTE:CHSPectrum:MARKer:SHOW:STATe	Determines whether to show or hide the readout for the selected marker.
DISPlay:LTE:CHSPectrum:RESet:SCALe	Resets the horizontal and vertical scale to default values.
DISPlay:LTE:CHSPectrum:WINDow:TRACe:GRATICule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:LTE:CHSPectrum:X:SCALe	Sets or queries the horizontal range.
DISPlay:LTE:CHSPectrum:X:SCALe:AUTO	Rescales the horizontal axis automatically to fit the waveform to the screen.

Table continued...

Command	Description
DISPlay:LTE:CHSPepectrum:X:SCALe:OFFSet	Sets or queries the center frequency of the trace.
DISPlay:LTE:CHSPepectrum:Y:SCALe	Sets or queries the vertical range of the graph.
DISPlay:LTE:CHSPepectrum:Y:SCALe:AUTO	Rescales the vertical scale automatically to fit the waveform on the screen.
DISPlay:LTE:CHSPepectrum:Y:SCALe:OFFSet	Sets or queries the vertical offset of the trace.
DISPlay:LTE:CONSte:SCALe	Sets or queries the scale (zoom).
DISPlay:LTE:CONSte:X:OFFSet	Sets or queries the horizontal offset.
DISPlay:LTE:CONSte:Y:OFFSet	Sets or queries the vertical offset.
DISPlay:LTE:MEASview<y>:DELete	Deletes the selected LTE measurement display in the application.
DISPlay:LTE:MEASview<y>:NEW	Opens a new LTE measurement display.
DISPlay:LTE:MEASview<y>:SELect	Selects an LTE measurement display in the application. The query returns the currently selected display.
DISPlay:LTE:PVTime:MARKer:SHOW:STATe	Shows or hides the marker readout.
DISPlay:LTE:PVTime:WINDow:TRACe:GRATicule:GRID:STATe	Determines whether to show or hide the graticule.
DISPlay:LTE:PVTime:X:SCALe	Sets or queries the vertical range.
DISPLay:LTE:PVTime:X:SCALe:AUTO	Rescales the horizontal axis automatically to fit the waveform to the screen.
DISPlay:LTE:PVTime:X:SCALe:RESet	Resets the horizontal scale to the default value for the time measurements.
DISPlay:LTE:PVTime:Y:SCALe	Sets or queries the horizontal range.
DISPlay:LTE:PVTime:Y:SCALe:AUTO	Rescales the vertical scale automatically to fit the waveform on the screen.
DISPlay:LTE:PVTime:Y:SCALe:OFFSet	Sets or queries the vertical offset (top edge of the vertical axis) of the trace.
DISPlay:LTE:PVTime:Y:SCALe:PDIVision	Sets or queries the value of the vertical scale in dB/division.
DISPlay:LTE:PVTime:Y:SCALe:RESet	Resets the vertical scale and position (offset) to the default value for the time measurements.

## Fetch commands

The FETCh commands retrieve the measurements from the data taken by the latest INITiate command. See the Index for commands not listed in the following table.

To perform a FETCh operation on fresh data, use the READ commands, which acquire a new input signal and fetch the measurement results from that data.

**Table 17: Fetch commands**

Command	Description
<b>FETCh TXGain command subgroup</b>	<b>Transmission Gain measurements</b>
FETCh:TXGain:AVERage?	Returns the mean value of the trace.

Table continued...



Command	Description
FETCh:TXGain:MAX?	Returns the maximum trace value.
FETCh:TXGain:MAXLocation?	Returns the frequency of the maximum trace value.
FETCh:TXGain:MIN?	Returns the minimum trace value.
FETCh:TXGain:MINLocation?	Returns the frequency of the minimum trace value.
FETCh:TXGain:TRACe<x>?	Returns the trace data of the selected trace.
<b>FETCh basic command subgroup</b>	<b>General fetch control</b>
FETCh:RFIN:IQ?	Returns time-domain IQ data for a specific acquisition data record.
<b>FETCh: BLUetooth subgroup</b>	<b>Bluetooth measurements</b>
FETCh:BIBEmissions:FTX?	Returns the band number corresponding to the transmitted FTX detected in the display.
FETCh:BIBEmissions:POWER?	Returns all channel power results.
FETCh:BIBEmissions:RESUlts:STATus?	Returns the Pass/Fail result status.
FETCh:BLUetooth:CONStE:FERRor?	Returns the frequency error.
FETCh:BLUetooth:CONStE:TRACe?	Queries the trace data.
FETCh:BLUetooth:EDlagram:FDEVIation?	Returns the frequency deviation versus time trace data with the X values.
FETCh:BLUetooth:EDlagram:FERRor?	Returns the frequency error.
FETCh:BLUetooth:EDlagram:HORiz:SCALe?	Returns the value of the horizontal scale.
FETCh:BLUetooth:EDlagram:I?	Returns the I versus time trace.
FETCh:BLUetooth:EDlagram:Q?	Returns the Q versus time trace.
FETCh:BLUetooth:FDVTime:TRACe?	Returns the trace data.
FETCh:BLUetooth:FREQuency:ERROR?	Returns the frequency error from the Bluetooth Constellation diagram.
FETCh:BLUetooth:FREQuency:ERROR:TYPE?	Returns the frequency error type in the Bluetooth Constellation display.
FETCh:BLUetooth:FREQuency:OFFSet:DRIFt:F1FZero?	Returns the drift in frequency offset ( $f_1-f_0$ —the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble) in the Bluetooth Frequency Deviation vs. Time display.
FETCh:BLUetooth:FREQuency:OFFSet:DRIFt:FNFN5?	Returns the maximum drift of the frequency offset in payload intervals spaced 50 $\mu$ s away in the Bluetooth Frequency Deviation vs. Time display.
FETCh:BLUetooth:FREQuency:OFFSet:DRIFt:FNFN5:INTERval?	Returns the interval at which the maximum drift of frequency offset in a duration of 50 $\mu$ s occurred.
FETCh:BLUetooth:FREQuency:OFFSet:DRIFt:FNFZERO?	Returns the maximum drift of the frequency offset of the intervals in the payload from the preamble packet in the Bluetooth Frequency Deviation vs. Time display.
FETCh:BLUetooth:FREQuency:OFFSet:DRIFt:FNFZERO:INTERval?	Returns the interval (n) at which the maximum drift $f_n-f_0$ occurred in the Bluetooth Frequency Deviation vs. Time display.

Table continued...

Command	Description
FETCh:BLUEtooth:FREQuency:OFFSet:MAX?	Returns the maximum frequency offset in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
FETCh:BLUEtooth:FREQuency:OFFSet:MAX:INTERval?	Returns the maximum frequency offset interval in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
FETCh:BLUEtooth:FREQuency:OFFSet:PREAmble?	Returns the offset calculated in the preamble region in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F?	Returns the average frequency value when the selected test pattern is Other in the Bluetooth Frequency Deviation vs. Time display.
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F2F1ratio?	Returns the average ratio value for the high and low frequency deviation in the Bluetooth Frequency deviation versus Time display.
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:FONE?	Returns the average frequency deviation for the low deviation pattern in the Bluetooth Frequency deviation versus Time display.
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:FTWO?	Returns the average frequency deviation for the high deviation pattern in the Bluetooth Frequency Deviation versus. Time display.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:COUNT?	Returns the number of packets analyzed to obtain average drift results.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:F1FZero?	Returns the drift $f_1-f_0$ .
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:F1FZero:STATus?	Returns the Pass/Fail drift $f_1-f_0$ result status in the packet.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:FNFN5?	Returns the maximum drift of the $f_n-f_{n-5}$ value.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:FNFN5:STATus?	Returns the Pass/Fail status of the maximum $f_n-f_{n-5}$ drift in the packet.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:FNFZero?	Returns the maximum drift $f_n-f_0$ .
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:DRIFt:FNFZero:STATus?	Returns the Pass/Fail result status of the $f_n-f_0$ maximum drift in the packet.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:MAX?	Returns the maximum frequency offset present in the packet.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:MAX:STATus?	Returns the Pass or Fail result status of the frequency offset maximum in the packet.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:PREAmble?	Returns the frequency offset value in the preamble region.
FETCh:BLUEtooth:SUMMmary:FREQuency:OFFSet:PREAmble:STATus?	Returns the Pass or Fail result status of the frequency offset in the preamble region.
FETCh:BLUEtooth:SUMMmary:MODUlation:CHARacteristics:AVErage:DELTA:F:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for $\Delta F$ average.
FETCh:BLUEtooth:SUMMmary:MODUlation:CHARacteristics:AVErage:DELTA:FONE:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for the $\Delta F_{1avg}$ .
FETCh:BLUEtooth:SUMMmary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentAge?	Returns the percentage of the $\Delta F_{2Max}$ values that are above the specified limit.
FETCh:BLUEtooth:SUMMmary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for the $\Delta F_{2Max\%}$ .

Table continued...

Command	Description
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentage:STATus?	Returns the Pass/Fail result status of the $\Delta F_{2Max\%}$ field.
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F?	Returns the $\Delta F_{avg}$ value.
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F2F1ratio?	Returns the ratio of the $\Delta F_{2avg}$ to $\Delta F_{1avg}$ .
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F2F1ratio:STATus?	Returns the Pass/Fail results status of the $\Delta F_{2avg}/\Delta F_{1avg}$ field.
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE?	Returns the $\Delta F_{1avg}$ value.
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE:STATus?	Returns the Pass/Fail results status of the $\Delta F_{1avg}$ .
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FTWO?	Returns the $\Delta F_{2avg}$ value.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:AVErage?	Returns the average output power.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:AVErage:STATus?	Returns the Pass/Fail results status of the average output power.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:COUNT?	Returns the number of packets analyzed out of 10 packets for the calculation of the output power.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:EDR:COUNT?	Returns the number of packets analyzed out of 10 packets for calculation of the output power for EDR.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:EDR:DPSK?	Returns the power in the DPSK portion of the EDR burst.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:EDR:GFSK?	Returns the power in the GFSK portion of the EDR burst.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK?	Returns the peak output power.
FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK:STATus?	Returns the Pass/Fail results status of the peak output power.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:CRc?	Returns the Basic Rate value of the CRC field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:ARQN?	Returns the value of ARQN field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:FLOW?	Returns the value of Flow field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:HEC?	Returns the value of HEC field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:LT:ADDRess?	Returns the LT address.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:SEQN?	Returns the value of SEQN field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:TYPE?	Returns the value of Type field.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:PAYLoad:LENGth?	Returns the Basic Rate payload length.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:PREAmble?	Returns the preamble of the Basic Rate packet.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:SYNC:WORD?	Returns the sync word.
FETCh:BLUEtooth:SUMMary:PACKet:BDR:TYPE?	Returns the packet type of the Basic Rate signal.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:ACCess:ADDRess?	Returns the 32-bit Access Address.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:CRc?	Returns the value of the CRC field.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:LENGth?	Returns the value of the length field.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:RX:ADDRess?	Returns the Rx address.

Table continued...

Command	Description
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:TX:ADDRess?	Returns the Tx address.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:TYPE?	Returns the PDU type.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PREAmble?	Returns the preamble of the BLE packet.
FETCh:BLUEtooth:SUMMary:PACKet:BLE:TYPE?	Returns the packet type.
<b>FETCH:NOISe subgroup</b>	<b>Noise Figure and Gain measurements</b>
FETCh:NOISe:GAIN:TRACe<x>:RESult?	This query returns the gain value in mdB of the specified trace in the Gain display.
FETCh:NOISe:TABLE:VALue?	Returns Noise Table data. A total of 7 values per frequency (28 bytes per frequency) is returned.
<b>FETCH:OBWidth subgroup</b>	<b>Bluetooth 20dB BW measurements</b>
FETCh:OBWidth:BOBW:XDBBandwidth?	Returns the difference between the higher and lower frequency points corresponding to the value where it is X dB less from the value at the center frequency.
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?	Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured inwards (coming from the ends to the center frequency).
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQuency?	Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the inward direction.
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQuency?	Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the inward direction.
FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth?	Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured outwards (from the center frequency to the ends).
FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency?	Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the outward direction.
FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency?	Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the outward direction.
<b>FETCH:P25 subgroup</b>	<b>P25 measurements</b>
FETCh:P25:CONSt:FERRor?	Returns the frequency error.
FETCh:P25:CONSt:TRACe?	Returns the display trace data.
FETCh:P25:EDIagram:FDEVIation?	Returns the Frequency Deviation versus Time trace data with the X value.
FETCh:P25:EDIagram:FERRor?	Returns the frequency error.
FETCh:P25:EDIagram:I?	Returns the I versus time trace data.
FETCh:P25:EDIagram:Q?	Returns the Q versus time trace data.
FETCh:P25:PVTime:BURSt:WIDTh?	Returns the value of the measured time width of the burst packet from power on to power down (or the end of waveform, if that occurs before power down), in seconds.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQuency:ACCUracy	Returns the Operational Frequency Accuracy measurement results.

Table continued...

Command	Description
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:ACCUracy:STATus	Returns the Operational Frequency Accuracy measurement status.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVIation:NEGPeak	Returns the Frequency Deviation, negative peak, measurement results.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVIation:NEGPeak:STATus	Returns the Status of Frequency Deviation, negative peak measurement status.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVIation:POSPeak	Returns the status of Frequency Deviation, positive peak results.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVIation:POSPeak:STATus	Returns the status of Frequency Deviation, positive peak status.
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQUency:ERror	Returns the Frequency Error measurement results.
FETCh:P25:SUMMary:MODUlation:MEASurement:SRA?	Returns the Symbol Rate Accuracy measurement results.
FETCh:P25:SUMMary:MODUlation:MEASurement:SRA:STATus	Returns the Symbol Rate Accuracy measurement status.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXOFF:STATus?	Returns HCPM Pmax-off measurement status.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXON:STATus?	Returns Status of HCPM Pmax-on measurement status.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXSs:STATus	Returns HCPM Pss-max measurement status.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MINImum:STATus?	Returns status of HCPM Pss-min measurement status based on the limit.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:OFFSlot:STATus?	Returns status of HCPM Offslot measurement based on the limit.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:HI	Returns P_ACP_HI value.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:LOW	Returns P_ACP_LOW value.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN	Returns Min Pk ACPR value.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN:STATus	Returns status of Min Pk ACPR value based on the limit.
FETCh:P25:SUMMary:POWEr:MEASurement:RF	Returns RF measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:RF:STATus	Returns Status of RF measurement status.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:ACQ:COUNT?	Returns the HCPM Time alignment's t_error_1 acquisition count over which the t_error_1 average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:STATus	Returns status of HCPM Time alignment's t_error_1 measurement results based on the limit.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ZERO:STATus?	Returns status of HCPM Time alignment's t_error_0 measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TIME:ENCODer:BI:STATus?	Returns status of Average Transmitter Power Encoder Attack Time,busy/idle, measurement status.

Table continued...

Command	Description
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:ENCODer:STATus?	Returns status of Average Transmitter Encoder Attack Time measurement results based on the limit.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:ENCODer?	Returns the Average Transmitter Encoder Attack Time measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:POWEr:BI:STATus	Returns status of Average Transmitter Power Attack Time, busy/idle, measurement results based on the limit.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:POWEr:BI?	Returns the Average Transmitter Power Attack Time, busy/idle, measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:POWEr:STATus	Returns status of Average Transmitter Power Attack Time measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:POWEr?	Returns the Average Transmitter Power Attack Time measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:THRUput: DELAY:ACQ:COUNT?	Returns the Throughput delay acquisition counts over which the Throughput delay average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:THRUput: DELAY:STATus	Returns status of Throughput delay measurement results based on the limit.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:THRUput: DELAY?	Returns the Throughput delay measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXOFF?	Returns the HCPM Pmax-off measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXON?	Returns the HCPM Pmax-on measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MAXSS?	Returns the HCPM Pss-max measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:MINImum?	Returns the HCPM Pss-min measurement results.
FETCh:P25:SUMMary:POWEr:MEASurement:HCPM:OFFSlot?	Returns the HCPM Offslot power results.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TERRor:ONE?	Returns the HCPM Time alignment's t_error_1 measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TERRor:ONE:ACQ:COUNT?	Returns the HCPM Time alignment's t_error_1 acquisition count over which the t_error_1 average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TERRor:ZERO?	Returns the HCPM Time alignment's t_error_0 measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TERRor:ZERO:ACQ:COUNT	Returns the HCPM Time alignment's t_error_0 acquisition count over which the t_error_0 average is done.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TOBSync?	Returns the HCPM Time alignment's t_obsync measurement results.
FETCh:P25:SUMMary:TRIGger:MEASurement:HCPM:TIME:ALIGN ment:TOBSync:ACQ:COUNT	Returns the HCPM Time alignment's t_obsync acquisition counts over which the t_obsync average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:ACQ:COUNT	Returns the Average Transmitter Power Attack acquisition count over which the t_obsync average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:ACQ:COUNT:BI?	Returns the Average Transmitter Power Attack, busy/idle, acquisition count over which the Busy/Idle Attack time average is computed.
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTack:TI ME:ENCODer?	Returns the Average Transmitter Encoder Attack Time measurement results.

Table continued...

Command	Description
FETCh:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI?	Returns the Average Transmitter Encoder Attack Time, busy/idle, measurement results.
FETCh:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER	Returns the Average Transmitter Power Attack Time measurement results.
FETCh:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI	Returns the Average Transmitter Power Attack Time, busy or idle, measurement results.
FETCh:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY	Returns the Throughput delay measurement results.
FETCh:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY:ACQ:COUNT	Returns the Throughput delay acquisition counts over which the Throughput delay average is computed.
<b>FETCh:RFIN subgroup</b>	<b>RF acquisition data</b>
FETCh:RFIN:IQ:HEADer?	Returns the header information for a specific acquisition data record.
FETCh:RFIN:IQ:SCALE?	Returns the scaling factor contained in the .tiq file header.
FETCh:RFIN:RECORD:IDS?	Returns the beginning and end ID numbers of acquisition data.
<b>FETCh:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
FETCh:ACPower?	Returns the ACPR measurement results.
FETCh:ACPower:CHANNEL:POWER?	Returns the average power of the main channel.
FETCh:ACPower:SPECTrum?	Returns spectrum trace data of the ACPR measurement.
FETCh:FM:PNegative?	Returns the negative peak frequency deviation (-Pk).
FETCh:FM:PPositive?	Returns the positive peak frequency deviation (+Pk).
FETCh:FM:PTPeak?	Returns the peak-peak frequency deviation (Pk-Pk).
FETCh:FM:RESult?	Returns the FM measurement results.
FETCh:FM:RMS?	Returns the RMS frequency deviation.
FETCh:PM:PNegative?	Returns the negative peak phase deviation (-Pk).
FETCh:PM:PPositive?	Returns the positive peak phase deviation (+Pk).
FETCh:PM:PTPeak?	Returns the peak-peak phase deviation (Pk-Pk).
FETCh:PM:RESult?	Returns the PM measurement results.
FETCh:PM:RMS?	Returns the PM measurement results.
<b>FETCh:AUDIO subgroup</b>	<b>Audio measurement</b>
FETCh:AUDIO:FERRor?	Returns the carrier frequency error in the audio measurement.
FETCh:AUDIO:FREQuency?	Returns the audio frequency.
FETCh:AUDIO:HARMonic:COUNT?	Returns the number of harmonics in the audio spectrum measurement.
FETCh:AUDIO:HARMonic<x>:AMPLitude?	Returns the amplitude of the specified harmonic in the audio spectrum measurement.
FETCh:AUDIO:HARMonic<x>:FREQuency?	Returns the frequency of the specified harmonic in the audio spectrum measurement.
FETCh:AUDIO:HNOise?	Returns the difference between the current RMS modulation value and the reference value.

Table continued...

Command	Description
FETCh:AUDio:HNREference?	Returns the Hum and Noise RMS modulation reference value.
FETCh:AUDio:HPTPeak?	Returns the half peak-to-peak modulation excursion.
FETCh:AUDio:MODDist?	Returns the modulation distortion for the audio measurement.
FETCh:AUDio:NHARmonic:COUNT?	Returns the number of non-harmonics in the audio spectrum measurement.
FETCh:AUDio:NHARmonic<x>:AMPLitude?	Returns the amplitude of the specified non-harmonic in the audio spectrum measurement.
FETCh:AUDio:NHARmonic<x>:FREQUENCY?	Returns the frequency of the specified non-harmonic in the audio spectrum measurement.
FETCh:AUDio:PNEGative?	Returns the minus peak modulation excursion.
FETCh:AUDio:POWer?	Returns the carrier power or signal power depending the signal type.
FETCh:AUDio:PPOSitive?	Returns the positive peak modulation excursion.
FETCh:AUDio:RMS?	Returns the RMS modulation excursion for the audio measurement.
FETCh:AUDio:SINad?	Returns the signal-to-noise and distortion for the audio measurement.
FETCh:AUDio:SNOise?	Returns the signal level to noise level with the harmonic distortion and non-harmonic distortion components removed.
FETCh:AUDio:SPECTrum:TRACe<x>?	Returns the audio spectrum trace data for the audio measurement.
FETCh:AUDio:THDist?	Returns the percentage of the total harmonic distortion.
FETCh:AUDio:THDist:DB?	Returns the total harmonic distortion in dB in the audio measurement.
FETCh:AUDio:TNHDist?	Returns the percentage of the total harmonic distortion in the audio measurement.
FETCh:AUDio:TNHDist:DB?	Returns the total non-harmonic distortion in dB in the audio measurement.
FETCh:AUDio:TNHDist?	Returns the percentage of the total non-harmonic distortion in the audio measurement.
<b>FETCh:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
FETCh:AVTime[:MEASview<y>]:AVERage?	Returns the RMS value.
FETCh:AVTime[:MEASview<y>]:{FIRSt SECond THIRd FOURth}?	Returns the trace data.
FETCh:AVTime[:MEASview<y>]:MAXLocation?	Returns the maximum value.
FETCh:AVTime[:MEASview<y>]:MAXLocation?	Returns the time at the maximum.
FETCh:AVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
FETCh:AVTime[:MEASview<y>]:MINLocation?	Returns the time at the minimum.
FETCh:AVTime[:MEASview<y>]:RESult?	Returns the measurement results.
<b>FETCh:CCDF subgroup</b>	<b>CCDF measurement</b>
FETCh:CCDF?	Returns the CCDF measurement results.
FETCh:CCDF:{FIRSt SECond THIRd}:X?	Returns the horizontal values of the specified trace.
FETCh:CCDF:{FIRSt SECond THIRd}:XY?	Returns the horizontal and vertical value pairs of the specified trace.

Table continued...



Command	Description
FETCh:CCDF:{FIRSt SECond THIRd}{:Y}?	Returns the vertical values of the specified trace.
<b>FETCh:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
FETCh:CONSte[:MEASview<x>]:FERRor?	Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency of the analyzer.
FETCh:CONSte[:MEASview<x>]:RESults?	Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.
FETCh:CONSte[:MEASview<x>]:TRACe?	Returns the constellation trace data.
<b>FETCh:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
FETCh:DDEMod[:MEASview<y>]:STABle?	Returns the symbol table data.
FETCh:DDEMod:SYNCh:WORD:LENGth?	Returns the length of the synch word in the symbol table.
FETCh:DDEMod:SYNCh:WORD:POSition?	Returns the position of the synch word in the symbol table.
<b>FETCh:DIQVtime subgroup (Option 21 only)</b>	<b>Demodulated I&amp;Q versus Time measurement</b>
FETCh:DIQVtime[:MEASview<y>]:FERRor?	Returns the frequency error.
FETCh:DIQVtime[:MEASview<y>]:I?	Returns the I versus Time trace data.
FETCh:DIQVtime[:MEASview<y>]:Q?	Returns the Q versus Time trace data.
<b>FETCh:DPX subgroup</b>	<b>DPX spectrum measurement</b>
FETCh:DPX:DDEnSity?	Returns the calculated density for the selected region in the DPXogram measurement.
FETCh:DPX:DGRam:LINE:COUnT?	Returns the number of lines in the DPXogram measurement.
FETCh:DPX:DGRam:TIME[:SCALE]:OFFSet?	Returns the time scale offset in the DPXogram measurement.
FETCh:DPX:RESults:TRACe<x>?	Returns waveform data in the DPX spectrum measurement.
FETCh:DPX:TDM:FREQuency:TXBer:BER?	Returns the Tx BER measurement BER results.
FETCh:DPX:TDM:FREQuency:TXBer:BITS?	Returns the number of bits counted in the active Tx BER measurement session.
FETCh:DPX:TDM:FREQuency:TXBer:ERRors?	Returns the number of errors counted in the active Tx BER measurement session.
FETCh:DPX:TDM:FREQuency:TXBer:STATe?	Returns the state of the Tx BER measurement.
FETCh:DPX:TRACe:AVERAge?	Returns waveform data of the average trace in the DPX spectrum measurement.
FETCh:DPX:TRACe:BITMap?	Returns trace waveform data of the bitmap trace.
FETCh:DPX:TRACe:MATH?	Returns waveform data of the math trace.
FETCh:DPX:TRACe:MAXimum?	Returns waveform data of the maximum trace.
FETCh:DPX:TRACe:MINimum?	Returns waveform data of the minimum trace.
<b>FETCh:EDlagram subgroup (Option 21 only)</b>	<b>Eye diagram</b>
FETCh:EDlagram[:MEASview<y>]:FDEVIation?	Returns the frequency deviation versus Time trace data with the X values.
FETCh:EDlagram[:MEASview<y>]:FERRor?	Returns the frequency error in the eye diagram measurement.

Table continued...

Command	Description
FETCh:EDlagram[:MEASview<y>]:I?	Returns the I versus Time trace data in the eye diagram measurement.
FETCh:EDlagram[:MEASview<y>]:Q?	Returns the Q versus Time trace data in the eye diagram measurement.
<b>FETCh:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
FETCh:EVM[:MEASview<y>]:FERRor?	Returns the frequency error in the EVM versus Time measurement.
FETCh:EVM[:MEASview<y>]:PEAK?	Returns the peak value in the EVM versus Time measurement.
FETCh:EVM[:MEASview<y>]:PINDeX?	Returns the time at the EVM peak.
FETCh:EVM[:MEASview<y>]:RMS?	Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.
FETCh:EVM[:MEASview<y>]:TRACe?	Returns the EVM versus Time trace data.
<b>FETCh:FDVTime subgroup (Option 21 only)</b>	<b>Frequency deviation versus Time measurement</b>
FETCh:FDVTime[:MEASview<y>]:FERRor?	Returns the frequency error in the Frequency deviation versus Time measurement.
FETCh:FDVTime[:MEASview<y>]:TRACe?	Returns the Frequency deviation versus Time trace data.
FETCh:{FSETtling PSETtling} subgroup (Option 12 only)	<b>Frequency and Phase Settling measurements</b>
FETCh:FSETtling:ERRor?	Returns the settled error in Hz in Frequency Settling.
FETCh:{FSETtling PSETtling}:FTTime?	Returns the settling time from the trigger position in seconds.
FETCh:{FSETtling PSETtling}:MASK[:PASS]?	Returns whether the input signal passes the mask test.
FETCh:{FSETtling PSETtling}:SETTled:FREQUency?	Returns the frequency at which the signal is considered settled.
FETCh:{FSETtling PSETtling}:SETTled[:PASS]?	Returns whether the input signal is settled with the tolerance range.
FETCh:{FSETtling PSETtling}:SETTled:TIME?	Returns the settled time in seconds.
FETCh:{FSETtling PSETtling}:SLMSd[:PASS]?	Returns whether the input signal is settled within the specified tolerance and the signal is settled longer than the Minimum Settled Duration.
FETCh:{FSETtling PSETtling}:STARt:TIME?	Returns the start time for the measurement in seconds.
FETCh:{FSETtling PSETtling}:TIME?	Returns the settling time in seconds. The settling time is measured from the start time.
FETCh:{FSETtling PSETtling}:TRACe<x>:X?	Returns the settling time in seconds. The settling time is measured from the start time.
FETCh:{FSETtling PSETtling}:TRACe<x>:XY?	Returns the time and frequency or phase value pairs of the specified trace.
FETCh:{FSETtling PSETtling}:TRACe<x>[:Y]?	Returns the frequency or phase values of the specified trace.
FETCh:{FSETtling PSETtling}:TRIGger:TIME?	Returns the time when the trigger occurred in seconds.
FETCh:{FSETtling PSETtling}:VALue?	Returns the settled value in Hz for Frequency Settling and in degrees for Phase Settling.
<b>FETCh:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
FETCh:FVTime[:MEASview<y>]?	Returns the Frequency versus Time trace data.
FETCh:FVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
FETCh:FVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the frequency drift is maximum.

Table continued...

Command	Description
FETCh:FVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
FETCh:FVTime[:MEASview<y>]:MINLocation?	Returns the time at which the frequency drift is minimum.
FETCh:FVTime[:MEASview<y>]:RESult?	Returns the measurement results.
<b>FETCh:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
FETCh:IQVTime[:MEASview<y>]:I?	Returns the I versus Time trace data.
FETCh:IQVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
FETCh:IQVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the I or Q level is maximum.
FETCh:IQVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
FETCh:IQVTime[:MEASview<y>]:MINLocation?	Returns the time at which the I or Q level is minimum.
FETCh:IQVTime[:MEASview<y>]:Q?	Returns the Q versus Time trace data.
FETCh:IQVTime[:MEASview<y>]:RESult?	Returns the measurement results.
<b>FETCh:MCPower subgroup</b>	<b>MCPR measurement</b>
FETCh:MCPower:ADJacent:CHANnels?	Returns the power of adjacent channels.
FETCh:MCPower:CHANnel:POWer?	Returns the reference power.
FETCh:MCPower:MAIN:CHANnels?	Returns the power of main channels.
FETCh:MCPower:SPEctrum?	Returns spectrum trace data.
<b>FETCh:MERRor subgroup (Option 21 only)</b>	<b>Magnitude error versus Time measurement</b>
FETCh:MERRor[:MEASview<y>]:FERRor?	Returns the frequency error in the Magnitude error versus Time measurement.
FETCh:MERRor[:MEASview<y>]:PEAK?	Returns the peak value in the Magnitude error versus Time measurement.
FETCh:MERRor[:MEASview<y>]:PINdex?	Returns the time at the Magnitude error peak.
FETCh:MERRor[:MEASview<y>]:RMS?	Returns the RMS value in the Magnitude error versus Time measurement.
FETCh:MERRor[:MEASview<y>]:TRACe?	Returns the Magnitude error versus Time trace data.
<b>FETCh:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
FETCh:OBWidth:BOBW:FREQuency:ERRor	Returns the frequency error.
FETCh:OBWidth:BOBW:RESUlts:STATus	Returns the test result as Pass or Fail.
FETCh:OBWidth:BOBW:XDBB:POWer	Returns the dB BW reference power.
FETCh:OBWidth:FREQuency:ERRor?	Returns the frequency error.
FETCh:OBWidth:OBWidth:BANdwidth?	Returns the occupied bandwidth.
FETCh:OBWidth:OBWidth:LEFT:FREQuency?	Returns the left (lower) frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:LEFT:LEVel?	Returns the level at the left frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:POWer?	Returns the reference power in the Occupied Bandwidth measurement.
FETCh:OBWidth:OBWidth:RIGHT:FREQuency?	Returns the right (higher) frequency of the occupied bandwidth.
FETCh:OBWidth:OBWidth:RIGHT:LEVel?	Returns the level at the right frequency of the occupied bandwidth.

Table continued...

Command	Description
FETCh:OBWidth:SPECTrum?	Returns spectrum trace data of the Occupied Bandwidth measurement.
FETCh:OBWidth:XDBBandwidth:BANDwidth?	Returns the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency?	Returns the left (lower) frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?	Returns the level at the left frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:POWEr?	Returns the reference power in the x dB bandwidth measurement.
FETCh:OBWidth:XDBBandwidth:RIGHT:FREQuency?	Returns the right (higher) frequency of the x dB bandwidth.
FETCh:OBWidth:XDBBandwidth:RIGHT:LEVel?	Returns the level at the right frequency of the x dB bandwidth.
<b>FETCh:OFDM subgroup</b>	<b>Orthogonal Frequency Division Multiplexing measurement</b>
FETCh:OFDM:APOWEr?	Returns the average power.
FETCh:OFDM:APOWEr:PEAK?	Returns the peak-to-average power.
FETCh:OFDM:CONStE:MAGNitude?	Returns the constellation magnitude data.
FETCh:OFDM:CONStE:PHASe?	Returns the constellation phase values.
FETCh:OFDM:CONStE:TYPE?	Returns the constellation context value.
FETCh:OFDM:CONStE:VALue?	Returns the constellation value.
FETCh:OFDM:CPE?	Returns the Common Pilot Error magnitude.
FETCh:OFDM:CRESPonse:MAGNitude?	Returns the channel response magnitude data.
FETCh:OFDM:CRESPonse:PHASe?	Returns the channel response phase data.
FETCh:OFDM:EVM:PEAK:DECibel:ALL?	Returns the peak EVM data for all subcarriers.
FETCh:OFDM:EVM:PEAK:DECibel:DATA?	Returns the peak EVM data for the data subcarriers.
FETCh:OFDM:EVM:PEAK:DECibel:PILots?	Returns the peak EVM data for the pilot subcarriers.
FETCh:OFDM:EVM:PEAK:PERCent:ALL?	Returns the peak EVM data for all subcarriers.
FETCh:OFDM:EVM:PEAK:PERCent:DATA?	Returns the peak EVM data for the data subcarriers.
FETCh:OFDM:EVM:PEAK:PERCent:PILots?	Returns the peak EVM data for the pilot subcarriers.
FETCh:OFDM:EVM:PEAK:SCARrier:ALL?	Returns the peak EVM data for all subcarriers.
FETCh:OFDM:EVM:PEAK:SCARrier:DATA?	Returns the peak EVM data for all data subcarriers.
FETCh:OFDM:EVM:PEAK:SCARrier:PILots?	Returns the peak EVM data for the pilot subcarriers.
FETCh:OFDM:EVM:PEAK:SYMBol:ALL?	Returns the peak EVM data for all subcarriers.
FETCh:OFDM:EVM:PEAK:SYMBol:DATA?	Returns the peak EVM data for the data subcarriers.
FETCh:OFDM:EVM:PEAK:SYMBol:PILots?	Returns the peak EVM data for the pilot subcarriers.
FETCh:OFDM:EVM:RMS:DECibel:ALL?	Returns the RMS EVM data for all subcarriers
FETCh:OFDM:EVM:RMS:DECibel:DATA?	Returns the RMS EVM data for the data subcarriers.
FETCh:OFDM:EVM:RMS:DECibel:PILots?	Returns the RMS EVM data for the pilot subcarriers.
FETCh:OFDM:EVM:RMS:PERCent:ALL?	Returns the peak RMS data for all subcarriers
FETCh:OFDM:EVM:RMS:PERCent:DATA?	Returns the peak RMS data for the data subcarriers.
FETCh:OFDM:EVM:RMS:PERCent:PILots?	Returns the peak RMS data for the pilot subcarriers.
FETCh:OFDM:EVM:TRACe<x>?	Returns the EVM trace data.

Table continued...

Command	Description
FETCh:OFDM:FERRor?	Returns the Frequency error reading.
FETCh:OFDM:FLATness:PASS?	Verifies the average power levels of the subcarriers remain within the limits defined for a particular standard.
FETCh:OFDM:FLATness:RESult?	Returns the summary results of each segment.
FETCh:OFDM:FLATness:TRACe<x>?	Returns the OFDM Spectral Flatness trace data.
FETCh:OFDM:GAIN:IMBalance?	Returns the gain imbalance.
FETCh:OFDM:IQ:ORIGin:OFFSet?	Returns the IQ origin offset.
FETCh:OFDM:MERRor:TRACe<x>?	Returns the magnitude error trace data.
FETCh:OFDM:PACKet:DIRectiOn?	Returns the direction of the packet.
FETCh:OFDM:PERRor:TRACe<x>?	Returns the phase error trace data.
FETCh:OFDM:POWer:TRACe<x>?	Returns the power trace data.
FETCh:OFDM:QUADrature:OFFSet?	Returns the quadrature offset.
FETCh:OFDM:SCARriers?	Returns the number of subcarriers.
FETCh:OFDM:SCARriers:SPACing?	Returns the subcarrier spacing.
FETCh:OFDM:STABLE:VALUe?	Returns the number of bytes in the symbol table.
FETCh:OFDM:SYMBol:CERRor?	Returns the symbol clock error.
FETCh:OFDM:SYMBol:COUNt?	Returns the number of symbols.
<b>FETCh:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
FETCh:PERRor[:MEASview<y>]:FERRor?	Returns the frequency error in the Phase error versus Time measurement.
FETCh:PERRor[:MEASview<y>]:PEAK?	Returns the peak value in the Phase error versus Time measurement.
FETCh:PERRor[:MEASview<y>]:PINDex?	Returns the time at the phase error peak.
FETCh:PERRor[:MEASview<y>]:RMS?	Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.
FETCh:PERRor[:MEASview<y>]:TRACe?	Returns the Phase error versus Time trace data.
<b>FETCh:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
FETCh:PHVTime[:MEASview<y>]?	Returns the Phase versus Time trace data.
FETCh:PHVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
FETCh:PHVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the phase is maximum.
FETCh:PHVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
FETCh:PHVTime[:MEASview<y>]:MINLocation?	Returns the time at which the phase is minimum.
FETCh:PHVTime[:MEASview<y>]:RESult?	Returns the results.
<b>FETCh:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
FETCh:PNOise:ALL?	Returns all the measurement results.
FETCh:PNOise:CARRier:FERRor?	Returns the carrier frequency error.
FETCh:PNOise:CARRier:POWer?	Returns the carrier power.
FETCh:PNOise:RESidual:FM?	Returns the residual FM.
Table continued...	

Command	Description
FETCh:PNOise:RMS:PNOise?	Returns the RMS phase noise.
FETCh:PNOise:SPECTrum<x>:X?	Returns the frequencies of the specified trace.
FETCh:PNOise:SPECTrum<x>:XY?	Returns the frequency and phase noise pairs of the specified trace.
FETCh:PNOise:SPECTrum<x>[:Y]?	Returns the phase noise values of the specified trace.
FETCh:PNOise:JITTer?	Returns the jitter.
<b>FETCh:PULSe subgroup</b>	<b>Pulsed RF measurements</b>
FETCh:PULSe[:RESult][:MEASview<y>]:ATX?	Returns the average transmitted power in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:AVERAge?	Returns the average on power in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:DRODb?	Returns the droop in dB in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:DROOp?	Returns the droop in the pulse table
FETCh:PULSe[:RESult][:MEASview<y>]:DUTPct?	Returns the duty factor (%) in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:DUTRatio?	Returns the duty factor (ratio) in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:FALL?	Returns the fall time in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:FDELta?	Returns the delta frequency in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:FRDeViation?	Returns the frequency deviation in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:IRAMplitude?	Returns the pulse impulse response amplitude measurement in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:IRTime?	Returns the impulse response time in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:MFRReqerror?	Returns the maximum frequency error in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:MPHerror?	Returns the maximum phase error in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:OVEDb?	Returns the overshoot in dB in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:OVERshoot?	Returns the overshoot in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:PHDeViation?	Returns the phase deviation in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:PPFRequency?	Returns the pulse-pulse carrier frequency in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:PPOWer?	Returns the peak power in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:PPPHase?	Returns the pulse-pulse carrier phase in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:RINterval?	Returns the repetition interval in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:RIPDb?	Returns the ripple in dB in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:RIPPlE?	Returns the ripple in the pulse table.
FETCh:PULSe[:RESult][:MEASview<y>]:RISE?	Returns the rise time in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:RMSFRreqerror?	Returns the RMS frequency error in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:RMSPherror?	Returns the RMS phase error in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:RRATE?	Returns the repetition rate in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:TIME?	Returns the time in the results table.
FETCh:PULSe[:RESult][:MEASview<y>]:WIDTh?	Returns the pulse width in the results table.
FETCh:PULSe:STATistics[:MEASview<y>]:ATX?	Returns the average transmitted power of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:AVERAge?	Returns the average on power of the statistics.

Table continued...

Command	Description
FETCh:PULSe:STATistics[:MEASview<y>]:DRODb?	Returns the droop in dB of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:DROOp?	Returns the droop of the statistics
FETCh:PULSe:STATistics[:MEASview<y>]:DUTPct?	Returns the duty factor (%) of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:DUTRatio?	Returns the duty factor (ratio) of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:FALL?	Returns the fall time of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:FDELta?	Returns the delta frequency measurement statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:FRDeviation?	Returns the frequency deviation of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:IRAMplitude?	Returns the Pulse Impulse Response amplitude measurement in the pulse statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:IRTime?	Returns the impulse response time in the pulse statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:MFRerror?	Returns the maximum frequency error of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:MPHerror?	Returns the maximum phase error of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:OVEDb?	Returns the overshoot measurement result in dB in the pulse statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:OVERshoot?	Returns the overshoot measurement result in the pulse statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:PHDeviation?	Returns the phase deviation of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:PPFRequency?	Returns the pulse-pulse carrier frequency of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:PPOwer?	Returns the peak power of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:PPPase?	Returns the pulse-pulse carrier phase of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RINterval?	Returns the repetition interval of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RIPDb?	Returns the ripple in dB of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RIPple?	Returns the ripple of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RISE?	Returns the rise time of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RMSFrequer?	Returns the RMS frequency error of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RMSPherror?	Returns the RMS phase error of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:RRATe?	Returns the repetition rate of the statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:WIDTh?	Returns the pulse width in the pulse statistics.
FETCh:PULSe:STATistics[:MEASview<y>]:X?	Returns the horizontal values of the trace data.
FETCh:PULSe:STATistics[:MEASview<y>]:XY?	Returns the horizontal and vertical values of the trace data.
FETCh:PULSe:STATistics[:MEASview<y>][:Y]?	Returns the vertical values of the trace data.
FETCh:PULSe:TRACe[:MEASview<y>]:X?	Returns the time values of the pulse trace.
FETCh:PULSe:TRACe[:MEASview<y>]:XY?	Returns the horizontal (time) and vertical value pairs of the pulse trace.
FETCh:PULSe:TRACe[:MEASview<y>][:Y]?	Returns the vertical values of the pulse trace.
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer?	Returns the outliers value exceeding the upper limit.
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer?	Returns the outliers value below the lower limit.

Table continued...

Command	Description
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULSe:COUNT?	Returns the number of pulses in the bin.
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULSe:TOtal?	Returns the total number of pulses.
FETCh:PULSe:CUMulative:STATistics[:MEASview<y>]	Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average and Standard Deviation values for the chosen measurement.
FETCh:PULSe:OGRAM[:MEASview<y>]:MAX:FRAME:COUNT	Returns the frame count value.
FETCh:PULSe:OGRAM[:MEASview<y>]:RBW	Returns the resolution bandwidth value only when Frequency vs Time display is opened in the Pulse-Ogram display.
FETCh:PULSe[:RESult][:MEASview<y>]:ADVanced	Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement.
FETCh:PULSe[:RESult][:MEASview<y>]:ADVanced:RANGE	Queries the Pulse count, Max, Max Time Stamp, Min, Min Time Stamp, Peak to Peak, Average, and Standard Deviation values for the chosen measurement in the Pulse Table display.
FETCh:PULSe[:RESult][:MEASview<y>]:FABS?	Returns the computed absolute frequency from the carrier.
FETCh:PULSe[:RESult][:MEASview<y>]:PPFD?	Returns the computed pulse to pulse frequency difference from the carrier.
FETCh:PULSe[:RESult][:MEASview<y>]:PPPD?	Returns the computed pulse to pulse phase difference from the carrier.
FETCh:PULSe:STATistics[:MEASview<y>]:ADVanced	Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement.
FETCh:PULSe:STATistics[:MEASview<y>]:FABS	Returns the absolute frequency.
FETCh:PULSe:STATistics[:MEASview<y>]:MAX:PULSe:NUMBER	Returns the maximum value at a certain pulse number.
FETCh:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMBER?	Returns the minimum value at a certain pulse number.
FETCh:PULSe:STATistics[:MEASview<y>]:PPFD	Returns the pulse-pulse frequency difference.
FETCh:PULSe:STATistics[:MEASview<y>]:PPPD	Returns the pulse-pulse phase difference.
<b>FETCh:[RF]:PRESelector subgroup</b>	<b>Preselector for RSA7100A</b>
FETCh:INPut[:RF]:PRESelector:STATe	Sets or queries the Preselector setting, which uses input filters for image suppression.
<b>FETCh:RFIN subgroup</b>	<b>All measurements</b>
FETCh:RFIN:IQ?	Returns time-domain IQ data for a specific acquisition data record.
FETCh:RFIN:IQ:HEADer?	Returns the header information for a specific acquisition data record.
FETCh:RFIN:IQ:SCALE?	Returns the internal RF linear data scaling factor contained in the .tiq file header
FETCh:RFIN:IQ:VHEader?	Returns the verbose header information for a specific acquisition data record.
FETCh:RFIN:RECORD:IDS?	Returns the beginning and end ID numbers of acquisition data.
<b>FETCh:SEM subgroup</b>	<b>Spectral Emissions Mask measurements</b>

Table continued...



Command	Description
FETCh:SEM:MHITs:COUNT?	Returns the number of rows in the results table
FETCh:SEM:MHITs<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified mask hit signal.
FETCh:SEM:MHITs<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified mask hit signal.
FETCh:SEM:MHITs<x>:FREQUency?	Returns the frequency of the specified mask hit signal.
FETCh:SEM:MHITs<x>:INTeg:ABSolute?	Returns the absolute integration amplitude of the specified mask hit signal.
FETCh:SEM:MHITs<x>:INTeg:RELative?	Returns the relative integration amplitude of the specified mask hit signal.
FETCh:SEM:MHITs<x>:MARGin:ABSolute?	Returns the absolute margin for the specified mask hit signal.
FETCh:SEM:MHITs<x>:MARGin:RELative?	Returns the relative margin for the specified mask hit signal.
FETCh:SEM:MHITs<x>:OFFSet?	Returns the offset for the specified mask hit signal.
FETCh:SEM:PASS?	Returns the pass/fail limit test result.
FETCh:SEM:REF:POWer?	Returns the reference power level.
FETCh:SEM:SPECtrum:X?	Returns the frequencies of the spectrum trace.
FETCh:SEM:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
FETCh:SEM:SPECtrum[:Y]?	Returns the amplitude of the spectrum trace.
<b>FETCh:SGRam subgroup</b>	<b>Spectrogram measurement</b>
FETCh:SGRam?	Returns the spectrogram trace data.
<b>FETCh:SIGNAlvu subgroup</b>	<b>Signal measurement</b>
FETCh:SIGNAlvu:ACQuisition:COUPLing	Returns the coupling value of the acquisition.
FETCh:SIGNAlvu:ACQuisition:MAX:RT:BW	Returns the maximum real time bandwidth value of the instrument.
FETCh:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:HIGHSide:MAXFreq?	Returns the maximum range of input center frequency in case of high injection side.
FETCh:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:HIGHSide:MINFreq?	Returns the minimum range of input center frequency in case of high injection side.
FETCh:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:LOWSide:MAXFreq?	Returns the maximum range of input center frequency in case of low injection side.
FETCh:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:LOWSide:MINFreq?	Returns the minimum range of input center frequency in case of low injection side.
FETCh:SIGNAlvu:ACQuisition:TERMination	Returns the termination value of the acquisition.
<b>FETCh:SPECtrum subgroup</b>	<b>Spectrum measurement</b>
FETCh:SPECtrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
<b>FETCh:SPURious subgroup</b>	<b>Spurious measurement</b>
FETCh:SPURious:CARRier:POWer?	Returns the carrier power.
FETCh:SPURious:COUNT?	Returns the number of spurious signals.
FETCh:SPURious:PASS?	Returns the pass/fail limit test result.
FETCh:SPURious:SPECtrum:X?	Returns the frequencies of the spectrum trace.
FETCh:SPURious:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
FETCh:SPURious:SPECtrum[:Y]?	Returns the amplitudes of the spectrum trace.

Table continued...

Command	Description
FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified spurious signal.
FETCh:SPURious:SPUR<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified spurious signal.
FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?	Returns the absolute frequency of the specified spurious signal.
FETCh:SPURious:SPUR<x>:FREQuency:RELative?	Returns the relative frequency of the specified spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:ABSolute?	Returns the absolute amplitude of the limit for a spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:RELative?	Returns the relative amplitude of the limit for a spurious signal.
FETCh:SPURious:SPUR<x>:LIMit:VIOlation?	Returns whether the specified spurious signal exceeds the limit or not.
FETCh:SPURious:SPUR<x>:RANGe?	Returns the frequency range in which the spurious signal occurred.
<b>FETCh:SQUality subgroup (Option 21 only)</b>	<b>Signal quality measurement</b>
FETCh:SQUality[:MEASview<y>]:FREQuency:DEVIation?	Returns the frequency deviation in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:FREQuency:DEVIation:TABLE?	Returns the number of columns and the values in the frequency deviation table for a signal quality measurement.
FETCh:SQUality[:MEASview<y>]:FREQuency:ERRor?	Returns the frequency error in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:GAIN:IMBalance?	Returns the gain imbalance in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:ORIGin:OFFSet?	Returns the origin offset in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM?	Returns the peak EVM (%) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB?	Returns the peak EVM (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet?	Returns the peak offset EVM (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation?	Returns the time at which the EVM is peak.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet?	Returns the time at which the offset EVM is peak.
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet?	Returns the peak offset EVM (%) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:FERRor?	Returns the peak FSK error in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:MERRor?	Returns the peak magnitude error (%) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:DB?	Returns the peak magnitude error (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:LOCation?	Returns the time at which the magnitude error is peak.
FETCh:SQUality[:MEASview<y>]:PEAK:PERRor?	Returns the peak phase error in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:PEAK:PERRor:LOCation?	Returns the time at which the phase error is peak.
FETCh:SQUality[:MEASview<y>]:QUADrature:ERRor?	Returns the quadrature error in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RHO?	Returns the $\rho$ (waveform quality) value in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:EVM?	Returns the RMS EVM (%) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB?	Returns the RMS EVM (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB:OFFSet?	Returns the RMS offset EVM (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:EVM:OFFSet?	Returns the RMS offset EVM (%) in the signal quality measurement.

Table continued...

Command	Description
FETCh:SQUality[:MEASview<y>]:RMS:FERRor?	Returns the RMS FSK error percent in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:MER:DB?	Returns the RMS MER (Modulation Error Ratio) in dB in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:MERRor?	Returns the RMS magnitude error (%) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:MERRor:DB?	Returns the RMS magnitude error (dB) in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:RMS:PERRor?	Returns the RMS phase error in the signal quality measurement.
FETCh:SQUality[:MEASview<y>]:SYMBOL:LENGth?	Returns the number of analyzed symbols.
FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE?	Returns the value of the calculated symbol rate in Hz.
FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE:ERRor?	Returns the value of the symbol rate error in percent (%).
<b>FETCh:TDIagram subgroup (Option 21 only)</b>	<b>Trellis diagram</b>
FETCh:TDIagram[:MEASview<y>]:FERRor?	Returns the frequency error.
FETCh:TDIagram[:MEASview<y>]:TRACe?	Returns the Trellis diagram trace data.
<b>FETCh:TOVerview subgroup</b>	<b>Time overview</b>
FETCh:TOVerview:MEASview<y>?	Returns the trace data.
<b>FETCh:WLAN subgroup</b>	<b>Wireless LAN measurements</b>
FETCh:WLAN:CONStE:MAGNitude?	Returns the magnitude data for the WLAN Constellation measurement.
FETCh:WLAN:CONStE:PHASe?	Returns the phase values for the WLAN Constellation measurement.
FETCh:WLAN:CONStE:TYPE?	Returns the type (context) for the WLAN Constellation measurement.
FETCh:WLAN:CONStE:VALue?	Returns the value (decoded symbol) for the WLAN Constellation measurement.
FETCh:WLAN:CRESPonse:MAGNitude?	Returns the magnitude data of the WLAN Channel Response measurement.
FETCh:WLAN:CRESPonse:PHASe?	Returns the phase data of the WLAN Channel Response measurement.
FETCh:WLAN:EVM:TRACe<x>?	Returns the trace data for the selected WLAN EVM trace.
FETCh:WLAN:FLATness:PASS?	Verifies the average power level values of the subcarriers remain within the limits defined for a particular standard.
FETCh:WLAN:FLATness:RESult?	Returns the summary results of each segment of the WLAN Spectral Flatness display.
FETCh:WLAN:FLATness:TRACe<x>?	Returns the WLAN Spectral Flatness trace data.
FETCh:WLAN:MERRor:TRACe<x>?	Returns the trace data for the selected WLAN Magnitude Error trace.
FETCh:WLAN:PERRor:TRACe<x>?	Returns the trace data for the selected WLAN Phase Error trace.
FETCh:WLAN:PVTime:BURSt:POWer?	Returns the average power of the burst packet.

Table continued...

Command	Description
FETCh:WLAN:PVTime:BURSt:WIDTh?	Returns the value of the measured time width of the burst packet from Power-On to Power-Down.
FETCh:WLAN:PVTime:FALL:TIME?	Returns the width of the falling edge.
FETCh:WLAN:PVTime:FALL:TIME:START?	Returns the start time of the falling edge.
FETCh:WLAN:PVTime:FALL:TIME:STOP?	Returns the stop time of the falling edge.
FETCh:WLAN:PVTime:FTIME?	Returns the width of the falling edge.
FETCh:WLAN:PVTime:RISE:TIME?	Returns the width of the rising edge.
FETCh:WLAN:PVTime:RISE:TIME:START?	Returns the start time of the rising edge.
FETCh:WLAN:PVTime:RISE:TIME:STOP?	Returns the stop time of the rising edge.
FETCh:WLAN:PVTime:RTIME?	Returns the width of the rising edge.
FETCh:WLAN:PVTime:TRACe:X?	Returns the horizontal values.
FETCh:WLAN:PVTime:TRACe:XY?	Returns the horizontal value (time in seconds) and vertical value (power).
FETCh:WLAN:PVTime:TRACe:Y?	Returns the vertical values (power).
FETCh:WLAN:STABLE:VALUe?	Returns the number of bytes in the symbol table.
FETCh:WLAN:SUMMary:BURSt:INDex?	Returns the index of the analyzed packet.
FETCh:WLAN:SUMMary:BURSt:POWer?	Returns the average power of all symbols in the packet.
FETCh:WLAN:SUMMary:BURSt:POWer:CFACTOR?	Returns the peak-to-average burst power factor.
FETCh:WLAN:SUMMary:CERRor?	Returns the symbol clock error.
FETCh:WLAN:SUMMary:CPE?	Returns the RMS magnitude error of the pilots.
FETCh:WLAN:SUMMary:EVM:PEAK:ALL?	Returns the peak EVM value for all subcarriers or all chips.
FETCh:WLAN:SUMMary:EVM:PEAK:CHIP:ALL?	Returns the chip at which the reported peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:CHIP: ONEKchips?	Returns the chip at which the reported peak EVM value over the first 1000 chips occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:DATA?	Returns the peak EVM value for the data subcarriers.
FETCh:WLAN:SUMMary:EVM:PEAK:ONEKchips?	Returns the peak EVM value measured over the first 1000 chips.
FETCh:WLAN:SUMMary:EVM:PEAK:PILOts?	Returns the peak EVM value for the pilots in the WLAN measurement.
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:ALL?	Returns the subcarrier on which the ALL peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:DATA?	Returns the subcarrier on which the DATA peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier: PILOts?	Returns the subcarrier on which the PILOT peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:ALL?	Returns the symbol on which the reported EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:DATA?	Returns the symbol on which the DATA peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:PILOts?	Returns the symbol on which the PILOT peak EVM value occurred.
FETCh:WLAN:SUMMary:EVM:RMS:ALL?	Returns the RMS EVM data for all subcarriers.
FETCh:WLAN:SUMMary:EVM:RMS:AVERAge?	Returns the average RMS EVM value.
FETCh:WLAN:SUMMary:EVM:RMS:DATA?	Returns the RMS EVM data for the data subcarriers.

Table continued...

Command	Description
FETCh:WLAN:SUMMary:EVM:RMS:MAXimum?	Returns the maximum RMS EVM value.
FETCh:WLAN:SUMMary:EVM:RMS:ONEKchips?	Returns the RMS EVM data for the 1k chips in the WLAN measurement measured over the first 1000 chips.
FETCh:WLAN:SUMMary:EVM:RMS:PILOts?	Returns the RMS EVM data for the pilot subcarriers.
FETCh:WLAN:SUMMary:FERRor?	Returns the difference between the measured carrier frequency of the signal and the measurement frequency setting.
FETCh:WLAN:SUMMary:HEADer?	Returns the received packet header information.
FETCh:WLAN:SUMMary:HTSig?	Returns the decimal values of the received packet HT-SIGNAL
FETCh:WLAN:SUMMary:IQOffset?	Returns the average magnitude of the DC subcarrier level relative to total signal power.
FETCh:WLAN:SUMMary:LSIG?	Returns the decimal values of the received packet (legacy) SIGNAL symbols fields.
FETCh:WLAN:SUMMary:PACKet?	Returns the type and number of symbols in the Preamble and Data portions of the packet.
FETCh:WLAN:SUMMary:PACKet:DMODulation?	Returns the modulation used in the Data symbols.
FETCh:WLAN:SUMMary:PACKet:FORMat?	Returns the packet format setting.
FETCh:WLAN:SUMMary:PACKet:GUARd?	Returns the packet guard interval used by the DATA symbols.
FETCh:WLAN:SUMMary:SIG?	Returns the decimal values of the received packet SIGNAL symbols fields.
FETCh:WLAN:SUMMary:VHTSig:A?	Returns the decimal values of the received packet VHT-SIGNAL(A) symbols fields.
FETCh:WLAN:SUMMary:VHTSig:B?	Returns the decimal values of the received packet VHT-SIGNAL(B) symbols fields.
<b>FETCh:LTE subgroup</b>	
FETCh:LTE:ACLR:MHITs:COUNT?	Returns the number of rows in the results table.
FETCh:LTE:ACLR:MHITs<x>:BANDwidth?	Returns the bandwidth value of the channel (row).
FETCh:LTE:ACLR:MHITs<x>:CHANnel:NAME?	Returns the channel name (row) of the specified index in the results table.
FETCh:LTE:ACLR:MHITs<x>:FREQuency:OFFSet?	Returns the frequency offset of the specified channel (row).
FETCh:LTE:ACLR:MHITs<x>:INTeg:LOWEr: ABSolute?	Returns the lower absolute value of the channel (row).
FETCh:LTE:ACLR:MHITs<x>:INTeg:LOWEr:RELative?	Returns the lower relative value of the channel (row).
FETCh:LTE:ACLR:MHITs<x>:INTeg:UPPEr:ABSolute?	Returns the upper absolute value of the channel (row).
FETCh:LTE:ACLR:MHITs<x>:INTeg:UPPEr:RELative?	Returns the upper relative value of the channel (row).
FETCh:LTE:ACLR:REFerence:POWEr?	Returns the reference power level.
FETCh:LTE:ACLR:RESUlts:STATus?	Returns the pass or fail status.
FETCh:LTE:ACLR:SPECTrum:X?	Returns the frequencies of the spectrum trace.
FETCh:LTE:ACLR:SPECTrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
FETCh:LTE:ACLR:SPECTrum:Y?	Returns the amplitude of the spectrum trace.
FETCh:LTE:CHSPectrum:OBW?	Returns the Occupied Bandwidth (OBW) in Hz.

Table continued...

Command	Description
FETCh:LTE:CHSPectrum:POWer:CHANnel?	Returns the Channel Power value.
FETCh:LTE:CHSPectrum:SPECTrum?	Returns spectrum trace data.
FETCh:LTE:CONSt:CELL:ID?	Queries the Cell ID value.
FETCh:LTE:CONSt:FREQuency:ERRor?	Queries Frequency Error value in Hz.
FETCh:LTE:CONSt:GROUp:ID?	Queries the Group ID value.
FETCh:LTE:CONSt:SECTor:ID?	Queries the Sector ID value.
FETCh:LTE:PVTime:OFFSlot:POWer?	Returns the TOff power measurement value in dBm/MHz.
FETCh:LTE:PVTime:RESUlts:STATus	Returns the pass or fail status.
FETCh:LTE:PVTime:TRACe:X?	Returns the horizontal values (time in seconds).
FETCh:LTE:PVTime:TRACe:XY?	Returns the horizontal value (time in seconds) and vertical value (power).
FETCh:LTE:PVTime:TRACe:Y?	Returns the vertical values (power).

## Initiate commands

Use the INITiate commands to control the acquisition of data.

**Table 18: Initiate commands**

Command	Description
INITiate:CONTInuous	Sets or queries whether to acquire data continuously.
INITiate[:IMMEdiate]	Starts input signal acquisition.
INITiate:RESume	Starts data acquisition.

## Input commands

Use the INPut commands to control the characteristics of the signal input.

**Table 19: Input commands**

Command	Description
INPut:CORRection:EXTernal:EDIT<x>:INTerpolation	Sets or queries the interpolation setting to use with the indicated external gain table.
INPut:CORRection:EXTernal:EDIT<x>:NEW	Creates the indicated external loss table.
INPut:CORRection:EXTernal:TYPE	Sets or queries the data type to use when applying the external loss table corrections.
INPut:{MLEVel RLEVel}	Sets or queries the reference level.
INPut[:RF]:ATTenuation	Sets or queries the input attenuation.
INPut[:RF]:ATTenuation:AUTO	Sets or queries whether to set the attenuation automatically.
INPut[:RF]:ATTenuation:MONitor:STATe	Sets or queries whether to enable to monitor attenuator use.

Table continued...

Command	Description
INPut[:RF]:GAIN:STATe (Option 50 or 51 only)	Sets or queries whether to enable the internal preamplifier.

## Mass memory commands

Use the MMEMory commands to manipulate files on the mass memory devices. See the Index for commands not listed in the following table.

For the trace specifier TRACe<x>, refer to `Trace Mnemonics`. (See page 2-124.)

Not all of these commands apply to SignalVu. See the individual command descriptions for details.

**Table 20: Mass memory (MMEMory) commands**

Command	Description
<b>MMEMory:{AM FM PM} subgroup</b>	<b>AM/FM/PM measurement</b>
MMEMory:{AM FM PM}:LOAD:TRACe	Loads trace data from the specified file.
<b>MMEMory:TXGain subgroup</b>	<b>Transmission Gain measurements</b>
MMEMory:TXGain:LOAD:TRACe<x>	Loads the specified saved trace from a file.
MMEMory:TXGain:SHOW:TRACe<x>	Displays a saved trace instead of a live trace.
MMEMory:TXGain:STORe:TRACe<x>	Saves the selected trace to a file for later recall and analysis.
<b>MMEMory:Noise subgroup</b>	<b>Noise Figure and Gain measurements</b>
MMEMory:NOISe:GAIN:LOAD:TRACe	Loads the Noise Gain trace data from the specified file.
MMEMory:NOISe:GAIN:SHOW:TRACe	Show or hide the Noise Gain trace from the specified file.
MMEMory:NOISe:GAIN:STORe:TRACe	Stores the Noise Gain trace data in the specified file.
MMEMory:Noise:LOAD:ENRCALibration	Loads the specified Noise Figure ENR table.
MMEMory:Noise:LOAD:FREQTABLE	Loads the specified Noise Figure Frequency table.
MMEMory:Noise:STORe:ENRCALibration	Saves the Noise Figure ENR table to the specified location.
MMEMory:Noise:STORe:FREQTABLE	Saves the Noise Figure Frequency table to the specified location.
MMEMory:{AM FM PM}:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:{AM FM PM}:STORe:TRACe	Stores trace data in the specified file.
<b>MMEMory:AUDio subgroup</b>	<b>Audio measurements</b>
MMEMory:AUDio:LOAD:FILTer	Loads a user-defined audio filter from the specified file.
<b>MMEMory:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
MMEMory:AVTime [:MEASview<y>]:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:AVTime [:MEASview<y>]:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:AVTime [:MEASview<y>]:STORe:TRACe<x>	Stores trace data in the specified file.
<b>MMEMory:CALibration subgroup</b>	<b>Measurement calibration</b>
MMEMory:CALibration:LOAD:CORRection: EXTernal:EDIT<x>	Loads the external loss table data from a specified file.
MMEMory:CALibration:STORe:CORRection: EXTernal:EDIT<x>	Stores an external loss table to a specified file.

Table continued...

Command	Description
<b>MMEMory:CCDF subgroup</b>	<b>CCDF measurement</b>
MMEMory:CCDF:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:CCDF:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:CCDF:STORe:TRACe<x>	Stores trace data in the specified file.
<b>MMEMory:DDEMod subgroup</b>	<b>DPX spectrum measurement</b>
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer: MEASurement:USER<y>	Loads a user-defined measurement filter from the specified file.
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer: MEASurement:UOTHer	Loads a user-defined measurement filter from the specified file.
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFerence: USER<y>	Loads the specified user-defined reference filter.
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFerence: UOTHer	Loads the specified user-defined reference filter
MMEMory:DDEMod[:SOURce<x>]:LOAD:SYMBOL:MAP	Loads the specified symbol map filename.
<b>MMEMory:DPX subgroup</b>	<b>DPX spectrum measurement</b>
MMEMory:DPX:TDM:FREQuency:TXBer: LOAD:PATTern	Loads the Tx BER User data pattern from the specified file.
MMEMory:DPX:LOAD:TRACe<x>	Loads trace data from the specified file.
MMEMory:DPX:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:DPX:STORe:TRACe<x>	Stores trace data in the specified file.
<b>MMEMory:{FSETtling PSETtling} subgroup</b>	<b>Frequency and Phase Settling time measurements</b>
[MMEMory:{FSETtling PSETtling}:LOAD:TRACe<x>	Loads the Frequency or Phase Settling Time trace data from the specified file.
[MMEMory:{FSETtling PSETtling}:SHOW:TRACe<x>	Enables display of a recalled Frequency or Phase Settling trace file in Trace<x>.
[MMEMory:{FSETtling PSETtling}:STORe:TRACe<x>	Stores the Frequency or Phase Settling trace data in the specified file.
<b>MMEMory:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
MMEMory:FVTime[:MEASview<y>]:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:FVTime[:MEASview<y>]:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:FVTime[:MEASview<y>]:STORe:TRACe	Stores trace data in the specified file.
<b>MMEMory:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
MMEMory:IQVTime[:MEASview<y>]:LOAD:TRACe:I	Loads I trace data from the specified file.
MMEMory:IQVTime[:MEASview<y>]:LOAD:TRACe:Q	Loads Q trace data from the specified file.
MMEMory:IQVTime[:MEASview<y>]:SHOW:TRACe:I	Loads trace data from the specified file.
MMEMory:IQVTime[:MEASview<y>]:SHOW:TRACe<x>:Q	Enables display of a recalled trace file.
MMEMory:IQVTime[:MEASview<y>]:STORe:TRACe:I	Stores I trace data in the specified file.
MMEMory:IQVTime[:MEASview<y>]:STORe:TRACe:Q	Stores Q trace data in the specified file.
<b>MMEMory:LTE subgroup</b>	

Table continued...



Command	Description
MMEMory:LTE:ACLR:LOAD:TABLE	Loads the LTE ACLR table containing the limits for enabled ranges from the specified .csv file.
MMEMory:LTE:ACLR:STORE:TABLE	Stores the LTE ACLR table containing the limits for enabled ranges in a specified .csv file, allowing you to export the file into Microsoft Excel or other database system.
<b>MMEMory:LOAD subgroup</b>	<b>Load commands</b>
MMEMory:LOAD:IQ	Loads time-domain IQ waveform from a specified file.
MMEMory:LOAD:STATe	Loads the instrument setup from a specified file.
MMEMory:LOAD:TRACe	Loads the trace data from a specified file.
<b>MMEMory:OFDM CONSTe subgroup</b>	<b>OFDM Constellation commands</b>
MMEMory:OFDM:CONSTe:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:OFDM:CONSTe:SHOW:TRACe	Enables the display of a recalled trace file.
MMEMory:OFDM:CONSTe:STORE:TRACe	Stores trace data in the specified file.
<b>MMEMory:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
MMEMory:PHVTime[:MEASview<y>]:LOAD:TRACe	Loads trace data from the specified file.
MMEMory:PHVTime[:MEASview<y>]:SHOW:TRACe	Enables display of a recalled trace file.
MMEMory:PHVTime[:MEASview<y>]:STORE:TRACe	Stores trace data in the specified file.
<b>MMEMory:PNOise subgroup</b>	<b>Phase noise measurement</b>
MMEMory:PNOise:LOAD:TRACe<x>	Loads the Phase Noise trace data from the specified file.
MMEMory:PNOise:SHOW:TRACe<x>	Enables display of a recalled Phase Noise trace file.
MMEMory:PNOise:STORE:TRACe<x>	Stores the Phase Noise trace data in the specified file.
<b>MMEMory:SEM subgroup</b>	<b>Spectral Emissions Mask measurement</b>
MMEMory:SEM:LOAD:TABLE	Loads the SEM table from the specified file.
MMEMory:SEM:STORE:TABLE	Stores the SEM table in the specified file.
<b>MMEMory:SGRam subgroup</b>	<b>Spurious measurement</b>
MMEMory:SGRam [:MEASview<y>]:LOAD:TRACe	Loads the trace data from a specified file.
MMEMory:SGRam [:MEASview<y>]:SHOW:TRACe	Enables display of a recalled trace file.
MMEMory:SGRam [:MEASview<y>]:STORE:TRACe	Stores the Spectrogram trace data in the specified file.
<b>MMEMory:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
MMEMory:SPECTrum [:MEASview<y>]:LOAD:TRACe	Loads the trace data from a specified file.
MMEMory:SPECTrum [:MEASview<y>]:SHOW:TRACe<x>	Enables display of a recalled trace file.
MMEMory:SPECTrum [:MEASview<y>]:STORE:TRACe<x>	Stores the spectrum trace data in the specified file.
<b>MMEMory:SPURious subgroup</b>	<b>Spurious measurement</b>
MMEMory:SPURious:LOAD:TABLE	Loads the spurious table from the specified file.
MMEMory:SPURious:STORE:TABLE	Stores the spurious table in the specified file.
<b>MMEMory:STORE subgroup</b>	<b>Store commands</b>
MMEMory:STORE:IQ	Saves time-domain IQ waveform in the acquisition memory to a specified file.

Table continued...

Command	Description
MMEMory:STORe:IQ:CSV	Saves time-domain IQ waveform in the acquisition memory to a specified file in the CSV (Comma Separated Values) format.
MMEMory:STORe:IQ:MAT	Saves time-domain IQ waveform in the acquisition memory to a specified file in the MATLAB format.
MMEMory:STORe:IQ:SELEct:DATA	Sets or queries the type of data stored in an acquisition data file.
MMEMory:STORe:IQ:SELEct:FRAMes	Sets or queries which frames to save when saving TIQ acquisition data files.
MMEMory:STORe:IQ:SELEct:LENGth	Sets or queries what portion of each record is saved when saving acquisition data.
MMEMory:STORe:MState	Stores the measurement parameters in a specified file.
MMEMory:STORe:RESults	Stores the measurement results including measurement parameters and trace data to a specified file.
MMEMory:STORe:SCReem	Stores the measurement results in a specified file.
MMEMory:STORe:STATe	Stores the instrument setup in a specified file.
MMEMory:STORe:TRACe	Stores trace data in a specified file.
<b>MMEMory:TOVerview subgroup</b>	<b>Spurious measurement</b>
MMEMory:TOVerview [:MEASview<y>]:LOAD:TRACe1	Loads the trace data from a specified file.
MMEMory:TOVerview [:MEASview<y>]:SHOW:TRACe1	Enables display of a recalled trace file.
MMEMory:TOVerview [:MEASview<y>]:STORe:TRACe1	Stores the selected trace data into the specified file.
<b>MMEMory:WLAN subbroup</b>	<b>Wireless LAN measurements</b>
MMEMory:WLAN:CRESpone:LOAD: TRACe<x>	Loads the selected WLAN Channel Response trace data from the specified file.
MMEMory:WLAN:CRESpone:SHOW: TRACe<x>	Enables the display of a recalled WLAN Channel Response trace.
MMEMory:WLAN:CRESpone:STORe: TRACe<x>	Stores the WLAN Channel Response measurement trace data in the specified file.

## Specifying the File

For loading and storing a file, specify the file following these rules

- You can omit the file extension to load and store data. The measurement-specific extension is automatically added.
- You can use the absolute path to specify the file name. For example, specify the SAMPLE1 file in the My Documents folder on the C drive as "C:\My Documents\SAMPLE1".
- initially.

Once a file is saved to a different directory, the new directory will be used as the default for all load and store operations.

## Output commands

Use the OUTPut commands to control the characteristics of the signal output. See the Index for commands not listed in the following table.

**Table 21: Output commands**

Command		Description
OUTPut:IF:{BANDwidth BWIDth}	(Option 05 only)	Sets or queries the IF output filter.
OUTPut:IF[:STATe]	(Option 05 only)	Sets or queries whether to turn on or off IF output.
OUTPut:IQ:SOURce	(Option 05 only)	Sets or queries the IQ data transfer via the LVDS card.
OUTPut:IQ[:STATe]	(Option 05 only)	Sets or queries whether to turn on or off IQ output.
OUTPut:NOISe[:STATe]		Sets or queries whether to turn on or off 28 V DC power.
OUTPut:ZS:{BANDwidth BWIDth}		Sets or queries the resolution bandwidth RBW filter.
OUTPut:ZS:{BANDwidth BWIDth}:AUTO		Sets or queries whether the resolution bandwidth will be set manually or automatically.
OUTPut:ZS:{BANDwidth BWIDth}: STATe		Sets or queries the state of the resolution bandwidth filter.
OUTPut:ZS[:STATe]		Enable or disable the Zero Span output.
OUTPut:ZS:DELay?		Queries the delay from the input connector to the Zero Span output.
OUTPut:ZS:FILTer:STATe		Sets or queries the signal path of the filter to the Zero Span output.
OUTPut:ZS:VOLTage		Sets or queries the voltage level at the Zero Span output.
OUTPut:ZS:VOLTage:AUTO		Sets or queries or queries how the output voltage is referenced.
OUTPut:TGen:BRIDge:MODE	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator bridge mode.
OUTPut:TGen:ENABle	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the power state of the tracking generator.
OUTPut:TGen:FREQuency	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator frequency in Hz.
OUTPut:TGen:INSTalled?	(Option 04 only; available for the RSA500A/600A series only))	Queries for the presence of the tracking generator hardware.
OUTPut:TGen:LEVel	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator output level in dBm.
OUTPut:IF:{BANDwidth BWIDth}	(Option 05 only)	Sets or queries the IF output filter.
OUTPut:IF[:STATe]	(Option 05 only)	Sets or queries whether to turn on or off IF output.
OUTPut:IQ:SOURce	(Option 05 only)	Sets or queries the IQ data transfer via the LVDS card.

Table continued...

Command		Description
OUTPut:IQ[:STATe]	(Option 05 only)	Sets or queries whether to turn on or off IQ output.
OUTPut:NOISe[:STATe]		Sets or queries whether to turn on or off 28 V DC power.
OUTPut:ZS:{BANDwidth BWIDth}		Sets or queries the resolution bandwidth RBW filter.
OUTPut:ZS:{BANDwidth BWIDth}:AUTO		Sets or queries whether the resolution bandwidth will be set manually or automatically.
OUTPut:ZS:{BANDwidth BWIDth}:STATe		Sets or queries the state of the resolution bandwidth filter.
OUTPut:ZS[:STATe]		Enable or disable the Zero Span output.
OUTPut:ZS:DELay?		Queries the delay from the input connector to the Zero Span output.
OUTPut:ZS:FILTer:STATe		Sets or queries the signal path of the filter to the Zero Span output.
OUTPut:ZS:VOLTagE		Sets or queries the voltage level at the Zero Span output.
OUTPut:ZS:VOLTagE:AUTO		Sets or queries or queries how the output voltage is referenced.
OUTPut:TGen:BRIDge:MODE	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator bridge mode.
OUTPut:TGen:ENABle	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the power state of the tracking generator.
OUTPut:TGen:FREQuency	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator frequency in Hz.
OUTPut:TGen:INSTAlled?	(Option 04 only; available for the RSA500A/600A series only))	Queries for the presence of the tracking generator hardware.
OUTPut:TGen:LEVel	(Option 04 only; available for the RSA500A/600A series only))	Sets or queries the tracking generator output level in dBm.

## Read commands

The READ commands acquire an input signal once in the single mode and obtain the measurement results from that data. See the Index for commands not listed in the following table.

To fetch the measurement results from the data currently residing in the memory without acquiring the input signal, use the FETCH commands.

**Table 22: Read commands**

Command	Description
READ:ACPower subgroup	Channel power and ACPR measurement
Table continued...	

Command	Description
READ:ACPower?	Returns the ACPR measurement results.
<b>READ:PULSe subgroup</b>	<b>Pulse measurements</b>
READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer?	Returns the outliers value exceeding the upper limit.
READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer?	Returns the outliers value below the lower limit.
READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULSe:COU Nt?	Returns the number of pulses in the bin.
READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULSe:TOT al?	Returns the total number of pulses.
READ:PULSe:CUMulative:STATistics[:MEASview<y>]	Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average and Standard Deviation values for the chosen measurement.
READ:PULSe[:RESult[:MEASview<y>]:ADVanced	Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in pulse table.
READ:PULSe[:RESult[:MEASview<y>]:ADVanced:RANGe	Queries the Pulse Count, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the chosen measurement in the Pulse Table display.
READ:PULSe[:RESult[:MEASview<y>]:FABS?	Returns the computed absolute frequency from the carrier in the pulse table.
READ:PULSe[:RESult[:MEASview<y>]:PPFD?	Returns the computed pulse to pulse frequency difference from the carrier in the pulse table.
READ:PULSe[:RESult[:MEASview<y>]:PPPD?	Returns the computed pulse to pulse phase difference from the carrier in the pulse table.
READ:PULSe:STATistics[:MEASview<y>]:ADVanced	Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:FABS	Returns the absolute frequency in pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:MAX:PULSe:NUMber	Returns the maximum value at a certain pulse number in pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMber?	Returns the minimum value at a certain pulse number in pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:PPFD	Returns the pulse-pulse frequency difference in pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:PPPD	Returns the pulse-pulse phase difference in pulse statistics.
<b>READ:Bluetooth subgroup</b>	<b>Bluetooth measurements</b>
READ:BIBEmissions:FTX?	Returns the band number corresponding to the transmitted FTX detected in the display.
READ:BIBEmissions:POWer?	Returns all channel power results.
READ:BIBEmissions:RESUltS:STATus?	Returns the Pass/Fail result status.
READ:BLUEtooth:CONStE:FERRor?	Queries the frequency error.
READ:BLUEtooth:CONStE:TRACe?	Returns the trace data for the display.
Table continued...	

Command	Description
READ:BLUEtooth:EDIagram:FDEVIation?	Returns the frequency deviation vs. time trace data with the X values.
READ:BLUEtooth:EDIagram:FERRor?	Returns the frequency error.
READ:BLUEtooth:EDIagram:HORiz:SCALe?	Returns the value of the horizontal scale.
READ:BLUEtooth:EDIagram:I?	Returns the I vs. time trace.
READ:BLUEtooth:EDIagram:Q?	Returns the Q vs. time trace.
READ:BLUEtooth:FDVTime:TRACe?	Returns the trace data or the display.
READ:BLUEtooth:FREQuency:ERRor?	Returns the frequency error.
READ:BLUEtooth:FREQuency:ERRor:TYPE	Returns the frequency error type.
READ:BLUEtooth:FREQuency:OFFSet:DRIFt:F1FZero?	Returns the drift in frequency offset ( $f_1 - f_0$ — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble).
READ:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFN5?	Returns the maximum drift of the frequency offset in payload intervals spaced 50 $\mu$ s away.
READ:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFN5:INTERval?	Returns the interval at which the maximum drift of frequency offset in a duration of 50 $\mu$ s occurred.
READ:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO?	Returns the maximum drift of the frequency offset of the intervals in the payload from the preamble packet in the Bluetooth Frequency Deviation vs. Time display.
READ:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO:INTERval?	Returns the interval (n) at which the maximum drift $f_n - f_0$ occurred in the Bluetooth Frequency Deviation vs. Time display.
READ:BLUEtooth:FREQuency:OFFSet:MAX?	Returns the maximum frequency offset in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
READ:BLUEtooth:FREQuency:OFFSet:MAX:INTERval?	Returns the maximum frequency offset interval in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
READ:BLUEtooth:FREQuency:OFFSet:PREAmble?	Returns the offset calculated in the preamble region in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.
READ:BLUEtooth:MODUlation:CHARacteristics: AVErage:F?	Returns the average frequency value when the selected test pattern is Other in the Bluetooth Frequency Deviation vs. Time display.
READ:BLUEtooth:MODUlation:CHARacteristics: AVErage:F2F1ratio?	Returns the average ratio value for the high and low frequency deviation in the Bluetooth Frequency Deviation vs. Time display.
READ:BLUEtooth:MODUlation:CHARacteristics: AVErage:FONE?	Returns the average frequency deviation for the low deviation pattern in the Bluetooth Frequency Deviation vs. Time display.
READ:BLUEtooth:MODUlation:CHARacteristics: AVErage:FTWO?	Returns the average frequency deviation for the high deviation pattern.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:COUNT?	Returns the number of packets analyzed to obtain average drift results.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:F1FZero?	Returns the drift $f_1 - f_0$ value.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:F1FZero:STATus?	Returns the Pass/Fail drift $f_1 - f_0$ result status in the packet.

Table continued...

Command	Description
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:FNFN5?	Returns the maximum drift of the $f_n-f_{n-5}$ value.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:FNFN5:STATus?	Returns the Pass/Fail status of the maximum $f_n-f_{n-5}$ drift in the packet.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:FNFZero?	Returns the maximum drift $f_n-f_0$ .
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: DRIFt:FNFZero:STATus?	Returns the Pass/Fail result status of the $f_n-f_0$ maximum drift in the packet.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: MAX?	Returns the maximum frequency offset present in the packet.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: MAX:STATus?	Returns the Pass/Fail result status of the frequency offset maximum in the packet.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: PREAmble?	Returns the frequency offset value in the preamble region.
READ:BLUEtooth:SUMMary:FREQuency:OFFSet: PREAmble:STATus?	Returns the Pass/Fail result status of the frequency offset in the preamble region.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:F:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for $\Delta F_{avg}$ .
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:FONE:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for the $\Delta F_{1avg}$ .
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentAge?	Returns the percentage of the $\Delta F_{2max\%}$ values that are above the specified limit.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentAge:COUNT?	Returns the number of packets analyzed out of 10 packets for $\Delta F_{2avg}$ .
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentAge:STATus?	Returns the Pass/Fail result status of the $\Delta F_{2Max\%}$ field.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F?	Returns the $\Delta F_{avg}$ value.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F2F1ratio?	Returns the ratio of the $\Delta F_{2avg}$ to $\Delta F_{1avg}$ .
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:F2F1ratio:STATus?	Returns the Pass/Fail results status of the $\Delta F_{2avg}/\Delta F_{1avg}$ field.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE?	Returns the delta F1 average value.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE:STATus?	Returns the Pass/Fail results status of the delta F1 average.
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:FTWO?	Returns the delta F2 average value,
READ:BLUEtooth:SUMMary:OUTPut:POWEr: AVErage?	Returns the average output power.
READ:BLUEtooth:SUMMary:OUTPut:POWEr: AVErage:STATus?	Returns the Pass/Fail results status of the average output power.
READ:BLUEtooth:SUMMary:OUTPut:POWEr:COUNT?	Returns the number of packets analyzed out of 10 packets for the calculation of the output power.
READ:BLUEtooth:SUMMary:OUTPut:POWEr:EDR: COUNT?	Returns the number of packets analyzed out of 10 packets for calculation of the output power for EDR.
READ:BLUEtooth:SUMMary:OUTPut:POWEr:EDR: DPSK?	Returns the power in the DPSK portion of the EDR burst.
READ:BLUEtooth:SUMMary:OUTPut:POWEr:EDR: GPSK?	Returns the power in the GFSK portion of the EDR burst.

Table continued...

Command	Description
READ:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK?	Returns the peak output power.
READ:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK: STATus?	Returns the Pass/Fail results status of the peak output power.
READ:BLUEtooth:SUMMary:PACKet:BDR:CRc?	Returns the Basic Rate value of the CRC field.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer: ARQN?	Returns the value of ARQN field.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer: FLOW?	Returns the value of Flow field.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer: HEC?	Returns the value of HEC field.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer:LT: ADDRess?	Returns the LT address.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer: SEQN?	Returns the value of SEQN field.
READ:BLUEtooth:SUMMary:PACKet:BDR:HEADer: TYPE?	Returns the value of Type field.
READ:BLUEtooth:SUMMary:PACKet:BDR:PAYLoad: LENGth?	Returns the preamble of the Basic Rate payload length.
READ:BLUEtooth:SUMMary:PACKet:BDR: PREAmble?	Returns the preamble of the Basic Rate packet.
READ:BLUEtooth:SUMMary:PACKet:BDR:SYNC: WORD?	Returns the sync word.
READ:BLUEtooth:SUMMary:PACKet:BDR:TYPE?	Returns the packet type of the Basic Rate signal.
READ:BLUEtooth:SUMMary:PACKet:BLE:ACCEss: ADDRess?	Returns the 32-bit Access Address.
READ:BLUEtooth:SUMMary:PACKet:BLE:CRc?	Returns the value of the CRC field.
READ:BLUEtooth:SUMMary:PACKet:BLE:PDU: HEADer:LENGth?	Returns the value of the PDU length field.
READ:BLUEtooth:SUMMary:PACKet:BLE:PDU: HEADer:RX:ADDRess?	Returns the PDU Rx address.
READ:BLUEtooth:SUMMary:PACKet:BLE:PDU: HEADer:TX:ADDRess?	Returns the PDU Tx address.
READ:BLUEtooth:SUMMary:PACKet:BLE:PDU: HEADer:TYPE?	Returns the PDU packet type.
READ:BLUEtooth:SUMMary:PACKet:BLE:PREAmble?	Returns the preamble of the BLE packet.
READ:BLUEtooth:SUMMary:PACKet:BLE:TYPE?	Returns the packet type.
READ:NOISe:TABLE:VALue?	Returns the noise table data.
READ:OBWidth:BOBW:XDBBbandwidth?	Returns the difference between the higher and lower frequency points corresponding to the value where it is X dB less from the value at the center frequency.
READ:OBWidth:BOBW:XDBMbandwidth:IN: BANDwidth?	Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured inwards (coming from the ends to the center frequency).
READ:OBWidth:BOBW:XDBMbandwidth:IN:LEFT: FREQUency?	Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the inward direction.
READ:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT: FREQUency?	Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the inward direction.
READ:OBWidth:BOBW:XDBMbandwidth:OUT: BANDwidth?	Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured outwards (from the center frequency to the ends).
READ:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT: FREQUency?	Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the outward direction.

Table continued...



Command	Description
READ:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT: FREQUENCY?	Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the outward direction.
<b>READ:P25 subgroup</b>	<b>P25 measurements</b>
READ:P25:CONStE:FERRor?	Returns the frequency error.
READ:P25:SUMMary:LIMIt: SRA	Returns Limit of Symbol Rate Accuracy measurement results.
READ:P25:SUMMary:MODUlation:MEASurement:FIDelity:STATus	Returns the Modulation Fidelity measurement status.
READ:P25:SUMMary:MODUlation:MEASurement:FIDelity?	Returns the Modulation Fidelity measurement results.
READ:P25:SUMMary:MODUlation:MEASurement:FREQUency:ACCUracy?	Returns the Operational Frequency Accuracy measurement results.
READ:P25:SUMMary:MODUlation:MEASurement:FREQUency:ACCUracy:STATus	Returns the status of Frequency Accuracy measurement status.
READ:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVlation:NEGPeak:STATus	Returns the Status of Frequency Deviation, negative peak status.
READ:P25:SUMMary:MODUlation:MEASurement:FREQUency:DEVlation:POSPeak:STATus	Returns the status of Frequency Deviation, positive peak status.
READ:P25:SUMMary:MODUlation:MEASurement:FREQUency:ERRor	Returns the Frequency Error measurement results.
READ:P25:SUMMary:MODUlation:MEASurement: SRA:STATus	Returns Status of Symbol Rate Accuracy measurement status.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:MAXOFF:STATus?	Returns HCPM Pmax-off measurement status.
READ:P25:SUMMary:POWEr:MEASurement:HCPM: MAXOFF?	This command returns the HCPM Pmax-off measurement results.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:MAXON:STATus	Returns Status of HCPM Pmax-on measurement status.
READ:P25:SUMMary:POWEr:MEASurement:HCPM: MAXON?	This command returns the HCPM Pmax-on measurement results.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:MAXSs:STATus	Returns HCPM Pss-max measurement status.
READ:P25:SUMMary:POWEr:MEASurement:HCPM: MAXSs?	Returns the HCPM Pss-max measurement results.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:MINImum:STATus?	Returns status of HCPM Pss-min measurement results based on the limit.
READ:P25:SUMMary:POWEr:MEASurement:HCPM: MINImum?	Returns the HCPM Pss-min measurement results.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:OFFSlot:STATus?	Returns Status of HCPM Offslot measurement based on the limit.
READ:P25:SUMMary:POWEr:MEASurement:HCPM: OFFSlot?	Returns the HCPM Offslot power results.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:HI	Returns P_ACP_HI value which is displayed on P25 Summary display.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:LOW	Returns P_ACP_LOW value which is displayed on P25 Summary display.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN	Returns Min Pk ACPR value which is displayed on P25 Summary Display.
READ:P25:SUMMary:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN:STATus	Returns status of Min Pk ACPR value based on the limit.
Table continued...	

Command	Description
READ:P25:SUMMARY:POWER:MEASUREMENT:RF:STATUS	Returns status of RF measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:RF?	Returns the RF output power results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:ACQ:COUNT?	Returns the HCPM Time alignment's t_error_1 acquisition count over which the t_error_1 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:STATUS	Returns status of HCPM Time alignment's t_error_1 measurement results based on the limit.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?	Returns the HCPM Time alignment t_error_1 measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT?	Returns the HCPM Time alignment's t_error_0 acquisition count over which the t_error_0 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:STATUS	Returns status of HCPM Time alignment's t_error_0 measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO?	Returns the HCPM Time alignment's t_error_0 measurement results over which the t_error_0 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSYNC:ACQ:COUNT?	Returns the HCPM Time alignment's t_obsync acquisition counts over which the t_obsync averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSYNC?	Returns the HCPM Time alignment's t_obsync measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI?	Returns the Average Transmitter Power Attack time busy/idle acquisition count over which the Phase 1 Attack time averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT?	Returns the Average Transmitter Power Attack acquisition count over which the Phase 1 Attack time averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI:STATUS	Returns status of Average Transmitter Power Encoder Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI?	Returns the Average Transmitter Power Encoder Attack Time busy/idle measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS	Returns Status of Average Transmitter Encoder Attack Time measurement results based on the limit.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER?	Returns the Average Transmitter Encoder Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI:STATUS	Returns the Average Transmitter Encoder Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI?	Returns the Average Transmitter Power Attack Time, busy/idle measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:STATUS	Returns Status of Average Transmitter Power Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER?	Returns the Average Transmitter Power Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THROUGHPUT:DELAY:ACQ:COUNT?	Returns the Throughput delay acquisition counts over which the Throughput delay averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THROUGHPUT:DELAY:STATUS	Returns Status of Throughput delay measurement results based on the limit.

Table continued...

Command	Description
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THROUGHPUT:DELAY?	Returns the Throughput delay measurement results.
READ:P25:EDLGRAM:FDEVIATION?	Returns the Frequency Deviation vs Time trace data with the X values.
READ:P25:EDLGRAM:FERROR?	Returns the frequency error.
READ:P25:PVTIME:BURST:POWER?	Returns the average power of the burst packet, in dBm, for the trace.
READ:P25:PVTIME:FALL:TIME?	Queries the value of the width of the falling edge, in seconds.
READ:P25:PVTIME:FTIME?	Queries the value of the width of the falling edge, in seconds.
READ:P25:PVTIME:TRACE:X?	Returns the horizontal values (time in seconds) for the P25 Power vs. Time trace.
READ:P25:PVTIME:TRACE:XY?	Returns the horizontal value (time in seconds) and vertical value (power) for the P25 Power vs. Time trace <x(n)><y(n)> is the horizontal value and vertical value pair at the <i>n</i> th data point.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY	Returns the Modulation Fidelity measurement results.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY	Returns the Operational Frequency Accuracy measurement results.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK	Returns the Frequency Deviation, negative peak, measurement results.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPEAK	Returns the Frequency Deviation, positive peak, measurement results.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR	Returns the Frequency Error measurement results.
READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA	Returns the Symbol Rate Accuracy measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF	This command returns the HCPM Pmax-off measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON	This command returns the HCPM Pmax-on measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS	Returns the HCPM Pss-max measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM	Returns the HCPM Pss-min measurement results.
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT	Returns the HCPM Offslot power results.
READ:P25:SUMMARY:POWER:MEASUREMENT:RF	Returns the RF output power results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE	Returns the HCPM Time alignment t_error_1 measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:ACQ:COUNT	Returns the HCPM Time alignment's t_error_1 acquisition count over which the t_error_1 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO	Returns the HCPM Time alignment's t_error_0 measurement results over which the t_error_0 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT	Returns the HCPM Time alignment's t_error_0 acquisition count over which the t_error_0 averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSYNC	Returns the HCPM Time alignment's t_obsync measurement results.

Table continued...

Command	Description
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSYNC:ACQ:COUNT	Returns the HCPM Time alignment's $t_{obsync}$ acquisition counts over which the $t_{obsync}$ averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT	Returns the Average Transmitter Power Attack acquisition count over which the Phase 1 Attack time averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI	Returns the Average Transmitter Power Attack time busy/idle acquisition count over which the Phase 1 Attack time averaging is done.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER	Returns the Average Transmitter Encoder Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI	Returns the Average Transmitter Power Encoder Attack Time, busy/idle measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER	Returns the Average Transmitter Power Attack Time measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI	Returns the Average Transmitter Power Attack Time, busy/idle measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THROUGHPUT:DELAY	Returns the Throughput delay measurement results.
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THROUGHPUT:DELAY:ACQ:COUNT	Returns the Throughput delay acquisition counts over which the Throughput delay averaging is done.
<b>READ:ACPower subgroup (continued)</b>	
READ:ACPower:CHANNEL:POWER?	Returns the average power of the main channel.
READ:ACPower:SPECTRUM?	Returns spectrum trace data of the ACPR measurement.
<b>READ:{AM FM PM} subgroup (Option 21 only)</b>	<b>AM/FM/PM measurement</b>
READ:{AM FM PM}?	Returns the trace data.
READ:AM:AMINDEX?	Returns the modulation index .
READ:AM:AMNEGATIVE?	Returns the negative peak modulation factor (-AM).
READ:AM:AMPPOSITIVE?	Returns the positive peak modulation factor (+AM).
READ:AM:RESULT?	Returns the AM measurement results.
READ:FM:PHALF?	Returns the half peak-peak frequency deviation (Pk-Pk/2).
READ:{FM PM}:FERROR?	Returns the frequency error in the Frequency modulation and Phase modulation measurements.
READ:FM:PNEGATIVE?	Returns the negative peak frequency deviation (-Pk).
READ:FM:PPOSITIVE?	Returns the positive peak frequency deviation (+Pk).
READ:FM:PTPEAK?	Returns the peak-peak frequency deviation (Pk-Pk).
READ:FM:RESULT?	Returns the FM measurement results.
READ:FM:RMS?	Returns the RMS frequency deviation.
READ:PM:PNEGATIVE?	Returns the negative peak phase deviation (-Pk).
READ:PM:PPOSITIVE?	Returns the positive peak phase deviation (+Pk).
READ:PM:PTPEAK?	Returns the peak-peak phase deviation (Pk-Pk).
READ:PM:RESULT?	Returns the PM measurement results.
Table continued...	

Command	Description
READ:PM:RMS?	Returns the RMS phase deviation.
<b>READ:AUDio subgroup</b>	<b>Audio measurement</b>
READ:AUDio:FERRor?	Returns the carrier frequency error in the audio measurement.
READ:AUDio:FREQuency?	Returns the audio frequency.
READ:AUDio:HARMonic:COUNT?	Returns the number of harmonics in the audio spectrum measurement.
READ:AUDio:HARMonic<x>:AMPLitude?	Returns the amplitude of the specified harmonic in the audio spectrum measurement.
READ:AUDio:HARMonic<x>:FREQuency?	Returns the frequency of the specified harmonic in the audio spectrum measurement.
READ:AUDio:HNOise?	Returns the difference between the current RMS modulation value and the reference value.
READ:AUDio:HNREference?	Returns the Hum and Noise RMS modulation reference value.
READ:AUDio:HPTPeak?	Returns the half peak-to-peak modulation excursion.
READ:AUDio:MODDist?	Returns the modulation distortion for the audio measurement.
READ:AUDio:NHARmonic:COUNT?	Returns the number of non-harmonics in the audio spectrum measurement.
READ:AUDio:NHARmonic<x>:AMPLitude?	Returns the amplitude of the specified non-harmonic in the audio spectrum measurement.
READ:AUDio:NHARmonic<x>:FREQuency?	Returns the frequency of the specified non-harmonic in the audio spectrum measurement.
READ:AUDio:PNEGative?	Returns the minus peak modulation excursion.
READ:AUDio:POWEr?	Returns the carrier power or signal power depending the signal type.
READ:AUDio:PPOSitive?	Returns the positive peak modulation excursion.
READ:AUDio:RMS?	Returns the RMS modulation excursion for the audio measurement.
READ:AUDio:SINad?	Returns the signal-to-noise and distortion for the audio measurement.
READ:AUDio:SNOise?	Returns the signal level to noise level with the harmonic distortion and non-harmonic distortion components removed.
READ:AUDio:SPECTrum:TRACe<x>?	Returns the audio spectrum trace data for the audio measurement.
READ:AUDio:THDist:DB?	Returns the total harmonic distortion in dB in the audio measurement.
READ:AUDio:THDist?	Returns the percentage of the total harmonic distortion in the audio measurement.
READ:AUDio:TNHDist:DB?	Returns the total non-harmonic distortion in dB in the audio measurement.
READ:AUDio:TNHDist?	Returns the percentage of the total non-harmonic distortion in the audio measurement.
<b>READ:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
READ:AVTime [:MEASview<y>]:AVERage?	Returns the RMS value.

Table continued...

Command	Description
READ:AVTime [:MEASview<y>]:{FIRSt SECond THIRd FOURth}?	Returns the trace data.
READ:AVTime [:MEASview<y>]:MAXimum?	Returns the maximum value.
READ:AVTime [:MEASview<y>]:MAXLocation?	Returns the time at the maximum.
READ:AVTime [:MEASview<y>]:MINimum?	Returns the minimum value.
READ:AVTime [:MEASview<y>]:MINLocation?	Returns the time at the minimum.
READ:AVTime [:MEASview<y>]:RESult?	Returns the measurement results.
<b>READ:CCDF subgroup</b>	<b>CCDF measurement</b>
READ:CCDF?	Returns the CCDF measurement results.
READ:CCDF:{FIRSt SECond THIRd}:X?	Returns the horizontal values of the specified trace.
READ:CCDF:{FIRSt SECond THIRd}:XY?	Returns the horizontal and vertical value pairs of the specified trace.
READ:CCDF:{FIRSt SECond THIRd}:Y?	Returns the vertical values of the specified trace.
<b>READ:CONSte subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
READ:CONSte[:MEASview<x>]:FERRor?	Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user selected center frequency of the analyzer.
READ:CONSte[:MEASview<x>]:RESults?	Returns the constellation measurement results of EVM RMS, peak and location displayed on the bottom of the screen.
READ:CONSte[:MEASview<x>]:TRACe?	Returns the constellation trace data.
<b>READ:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
READ:DDEMod[:MEASview<y>]:STABle?	Returns the symbol table data.
<b>READ:DIQVtime subgroup (Option 21 only)</b>	<b>Demodulated I&amp;Q versus Time measurement</b>
READ:DIQVtime[:MEASview<y>]:FERRor?	Returns the frequency error.
READ:DIQVtime[:MEASview<y>]:I?	Returns the I versus Time trace data.
READ:DIQVtime[:MEASview<y>]:Q?	Returns the Q versus Time trace data.
<b>READ:DPX subgroup</b>	<b>DPX Spectrum measurement</b>
READ:DPX:DDENsity?	Returns the calculated (DPX) density for the selected region.
READ:DPX:DGRam:LINE:COUNT?	Returns the number of lines in the DPXogram measurement.
READ:DPX:DGRam:TIME[:SCALe]:OFFSet?	Returns the time scale offset in the DPXogram measurement.
READ:DPX:RESults:TRACe<x>?	Returns waveform data of specified trace.
READ:DPX:TRACe:AVERAge?	Returns waveform data of the average trace.
READ:DPX:TRACe:BITMap?	Returns trace waveform data of the bitmap trace.
READ:DPX:TRACe:MATH?	Returns waveform data of the math trace.
READ:DPX:TRACe:MAXimum?	Returns waveform data of the maximum trace.
READ:DPX:TRACe:MINimum?	Returns waveform data of the minimum trace.
<b>READ:EDlagram subgroup (Option 21 only)</b>	<b>Eye diagram</b>
READ:EDlagram[:MEASview<y>]:FDEVIation?	Returns the frequency deviation versus Time trace data with the X values.
READ:EDlagram[:MEASview<y>]:FERRor?	Returns the frequency error in the eye diagram measurement.

Table continued...

Command	Description
READ:EDlagram[:MEASview<y>]:I?	Returns the I versus Time trace data.
READ:EDlagram[:MEASview<y>]:Q?	Returns the Q versus Time trace data.
<b>READ:EVM subgroup (Option 21 only)</b>	<b>EVM versus Time measurement</b>
READ:EVM[:MEASview<y>]:FERRor?	Returns the frequency error in the EVM versus Time measurement.
READ:EVM[:MEASview<y>]:PEAK?	Returns the peak value.
READ:EVM[:MEASview<y>]:PINDeX?	Returns the time at the EVM peak.
READ:EVM[:MEASview<y>]:RMS?	Returns the RMS value.
READ:EVM[:MEASview<y>]:TRACe?	Returns the EVM versus Time trace data.
<b>READ:FDVTime subgroup (Option 21 only)</b>	<b>Frequency deviation versus Time measurement</b>
READ:FDVTime[:MEASview<y>]:FERRor?	Returns the frequency error in the Frequency deviation versus Time measurement.
READ:FDVTime[:MEASview<y>]:TRACe?	Returns the Frequency deviation versus Time trace data.
<b>READ:{FSETtling PSETtling} subgroup</b>	<b>Frequency and Phase Settling Time measurements</b>
READ:FSETtling:ERRor?	Returns the settled error in Hz in Frequency Settling.
READ:{FSETtling PSETtling}:FTTime?	Returns the settling time from the trigger position in seconds.
READ:{FSETtling PSETtling}:MASK[:PASS]?	Returns whether the input signal passes the mask test.
READ:{FSETtling PSETtling}:SETTled:FREQuency?	Returns the frequency at which the signal is considered settled.
READ:{FSETtling PSETtling}:SETTled[:PASS]?	Returns whether the input signal is settled with the tolerance range.
READ:{FSETtling PSETtling}:SETTled:TIME?	Returns the settled time in seconds.
READ:{FSETtling PSETtling}:SLMSd[:PASS]?	Returns whether the input signal is settled within the specified tolerance and the signal is settled longer than the Minimum Settled Duration.
READ:{FSETtling PSETtling}:STARt:TIME?	Returns the start time for the measurement in seconds.
READ:{FSETtling PSETtling}:TIME?	Returns the settling time in seconds.
READ:{FSETtling PSETtling}:TRACe<x>:X?	Returns the Frequency or Phase values of the specified trace.
READ:{FSETtling PSETtling}:TRACe<x>:XY?	Returns the time and frequency or phase value pairs of the specified trace.
READ:{FSETtling PSETtling}:TRACe<x>[:Y]?	Returns the frequency or phase values of the specified trace.
READ:{FSETtling PSETtling}:TRIGger:TIME?	Returns the time when the trigger occurred in seconds.
READ:{FSETtling PSETtling}:VALue?	Returns the settled value in Hz for Frequency Settling and in degrees for Phase Settling.
<b>READ:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
READ:FVTime[:MEASview<y>]?	Returns the Frequency versus Time trace data.
READ:FVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
READ:FVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the frequency drift is maximum.
READ:FVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
READ:FVTime[:MEASview<y>]:MINLocation?	Returns the time at which the frequency drift is minimum.
READ:FVTime[:MEASview<y>]:RESult?	Returns the measurement results.

Table continued...

Command	Description
<b>READ:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
READ:IQVTime[:MEASview<y>]:I?	Returns the I versus Time trace data.
READ:IQVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
READ:IQVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the I or Q level is maximum.
READ:IQVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
READ:IQVTime[:MEASview<y>]:MINLocation?	Returns the time at which the I or Q level is minimum.
READ:IQVTime[:MEASview<y>]:Q?	Returns the Q versus Time trace data.
READ:IQVTime[:MEASview<y>]:RESult?	Returns the measurement results.
<b>READ:MCPower subgroup</b>	<b>MCPR measurement</b>
READ:MCPower:ADJacent:CHANnels?	Returns the power of adjacent channels.
READ:MCPower:CHANnel:POWer?	Returns the reference power.
READ:MCPower:MAIN:CHANnels?	Returns the power of main channels.
READ:MCPower:SPECTrum?	Returns spectrum trace data.
<b>READ:MERRor subgroup (Option 21 only)</b>	<b>Magnitude Error versus Time measurement</b>
READ:MERRor[:MEASview<y>]:FERRor?	Returns the frequency error.
READ:MERRor[:MEASview<y>]:PEAK?	Returns the peak value.
READ:MERRor[:MEASview<y>]:PINDEX?	Returns the time at the Magnitude error peak.
READ:MERRor[:MEASview<y>]:RMS?	Returns the RMS (Root-Mean-Square) value.
READ:MERRor[:MEASview<y>]:TRACe?	Returns the Magnitude error versus Time trace data.
<b>READ:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
READ:OBWidth:FREQuency:ERRor?	Returns the frequency error.
READ:OBWidth:OBWidth:BANDwidth?	Returns the occupied bandwidth.
READ:OBWidth:OBWidth:LEFT:FREQuency?	Returns the left (lower) frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:LEFT:LEVel?	Returns the level at the left frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:POWer?	Returns the reference power in the Occupied Bandwidth measurement.
READ:OBWidth:OBWidth:RIGHT:FREQuency?	Returns the right (higher) frequency of the occupied bandwidth.
READ:OBWidth:OBWidth:RIGHT:LEVel?	Returns the level at the right frequency of the occupied bandwidth.
READ:OBWidth:SPECTrum?	Returns spectrum trace data of the Occupied Bandwidth measurement.
READ:OBWidth:XDBBandwidth:BANDwidth?	Returns the x dB bandwidth.
READ:OBWidth:XDBBandwidth:LEFT:FREQuency?	Returns the left (lower) frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:LEFT:LEVel?	Returns the level at the left frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:POWer?	Returns the reference power in the x dB bandwidth measurement.
READ:OBWidth:XDBBandwidth:RIGHT:FREQuency?	Returns the right (higher) frequency of the x dB bandwidth.
READ:OBWidth:XDBBandwidth:RIGHT:LEVel?	Returns the level at the right frequency of the x dB bandwidth.
<b>READ:OFDM subgroup</b>	<b>Orthogonal Frequency Division Multiplexing measurement</b>
READ:OFDM:APOWer?	Returns the average power.

Table continued...



Command	Description
READ:OFDM:APOWer:PEAK?	Returns the peak-to-average power.
READ:OFDM:CONSte:MAGNitude?	Returns the constellation magnitude data.
READ:OFDM:CONSte:PHASe?	Returns the constellation phase data.
READ:OFDM:CONSte:TYPE?	Returns the constellation context value
READ:OFDM:CONSte:VALue?	Returns the constellation value.
READ:OFDM:CPE?	Returns the Common Pilot Error magnitude.
READ:OFDM:CRESPonse:MAGNitude?	Returns the channel response magnitude data.
READ:OFDM:CRESPonse:PHASe?	Returns the channel response phase data.
READ:OFDM:EVM:PEAK:DECibel:ALL?	Returns the peak EVM data for all subcarriers.
READ:OFDM:EVM:PEAK:DECibel:DATA?	Returns the peak EVM data for the data subcarriers.
READ:OFDM:EVM:PEAK:DECibel:PILots?	Returns the peak EVM data for the pilot subcarriers.
READ:OFDM:EVM:PEAK:PERCent:ALL?	Returns the peak EVM data for all subcarriers.
READ:OFDM:EVM:PEAK:PERCent:DATA?	Returns the peak EVM data for the data subcarriers.
READ:OFDM:EVM:PEAK:PERCent:PILots?	Returns the peak EVM data for the pilot subcarriers.
READ:OFDM:EVM:PEAK:SCARrier:ALL?	Returns the peak EVM data for all subcarriers.
READ:OFDM:EVM:PEAK:SCARrier:DATA?	Returns the peak EVM data for the data subcarriers.
READ:OFDM:EVM:PEAK:SCARrier:PILots?	Returns the peak EVM data for the pilot subcarriers.
READ:OFDM:EVM:PEAK:SYMBol:ALL?	Returns the peak EVM data for all subcarriers.
READ:OFDM:EVM:PEAK:SYMBol:DATA?	Returns the peak EVM data for the data subcarriers
READ:OFDM:EVM:PEAK:SYMBol:PILots?	Returns the peak EVM data for the pilot subcarriers.
READ:OFDM:EVM:RMS:DECibel:ALL?	Returns the RMS EVM data for all subcarriers.
READ:OFDM:EVM:RMS:DECibel:DATA?	Returns the RMS EVM data for the data subcarriers.
READ:OFDM:EVM:RMS:DECibel:PILots?	Returns the RMS EVM data for the pilot subcarriers.
READ:OFDM:EVM:RMS:PERCent:ALL?	Returns the peak RMS data for all subcarriers.
READ:OFDM:EVM:RMS:PERCent:DATA?	Returns the peak RMS data for the data subcarriers.
READ:OFDM:EVM:RMS:PERCent:PILots?	Returns the peak RMS data for the pilot subcarriers.
READ:OFDM:EVM:TRACe<x>?	Returns the EVM trace data.
READ:OFDM:FERRor?	Returns the Frequency error reading.
READ:OFDM:FLATness:PASS?	Verifies the average power levels of the subcarriers remain within the limits defined for a particular standard.
READ:OFDM:FLATness:RESult?	Returns the summary results of each segment.
READ:OFDM:FLATness:TRACe<x>?	Returns the OFDM Spectral Flatness trace data.
READ:OFDM:GAIN:IMBalance?	Returns the gain imbalance.
READ:OFDM:IQ:ORIGin:OFFSet?	Returns the IQ origin offset.
READ:OFDM:MERRor:TRACe<x>?	Returns the magnitude error trace data.
READ:OFDM:PACKet:DIRection?	Returns the direction of the packet.
READ:OFDM:PERRor:TRACe<x>?	Returns the phase error trace data.

Table continued...

Command	Description
READ:OFDM:POWer:TRACe<x>?	Returns the power trace data.
READ:OFDM:QUADrature:OFFSet?	Returns the quadrature offset.
READ:OFDM:SCARriers?	Returns the number of subcarriers.
READ:OFDM:SCARriers:SPACing?	Returns the subcarrier spacing.
READ:OFDM:STABle:VALUe?	Returns the number of bytes in the symbol table.
READ:OFDM:SYMBol:CERRor?	Returns the symbol clock error.
READ:OFDM:SYMBol:COUNT?	Returns the number of symbols.
<b>READ:PERRor subgroup (Option 21 only)</b>	<b>Phase error versus Time measurement</b>
READ:PERRor[:MEASview<y>]:FERRor?	Returns the frequency error in the Phase error versus Time measurement.
READ:PERRor[:MEASview<y>]:PEAK?	Returns the peak value in the Phase error versus Time measurement.
READ:PERRor[:MEASview<y>]:PINDex?	Returns the time at the phase error peak.
READ:PERRor[:MEASview<y>]:RMS	Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.
READ:PERRor[:MEASview<y>]:TRACe?	Returns the Phase error versus Time trace data.
<b>READ:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
READ:PHVTime[:MEASview<y>]?	Returns the Phase versus Time trace data.
READ:PHVTime[:MEASview<y>]:MAXimum?	Returns the maximum value.
READ:PHVTime[:MEASview<y>]:MAXLocation?	Returns the time at which the phase is maximum.
READ:PHVTime[:MEASview<y>]:MINimum?	Returns the minimum value.
READ:PHVTime[:MEASview<y>]:MINLocation?	Returns the time at which the phase is minimum.
READ:PHVTime[:MEASview<y>]:RESult?	Returns the results.
<b>READ:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurements</b>
READ:PNOise:ALL?	Returns all the measurement results.
READ:PNOise:CARRier:FERRor?	Returns the carrier frequency error.
READ:PNOise:CARRier:POWer?	Returns the carrier power.
READ:PNOise:RESidual:FM?	Returns the residual FM.
READ:PNOise:RMS:PNOise?	Returns the RMS phase noise.
READ:PNOise:SPECTrum<x>:X?	Returns the frequencies of the specified trace.
READ:PNOise:SPECTrum<x>:XY?	Returns the frequency and phase noise pairs of the specified trace.
READ:PNOise:SPECTrum<x>[:Y]?	Returns the phase noise values of the specified trace.
READ:PNOise:JITTer?	Returns the jitter.
<b>READ:PULSe subgroup</b>	<b>Pulsed RF measurements</b>
READ:PULSe[:RESult[:MEASview<y>]:ATX?	Returns the average transmitted power in the results table.
READ:PULSe[:RESult[:MEASview<y>]:AVERAge?	Returns the average on power in the results table.
READ:PULSe[:RESult[:MEASview<y>]:DRODb?	Returns the droop in dB in the pulse table.
READ:PULSe[:RESult[:MEASview<y>]:DROOp?	Returns the droop in the pulse table.

Table continued...

Command	Description
READ:PULSe[:RESult][:MEASview<y>]:DUTPct?	Returns the duty factor (%) in the results table.
READ:PULSe[:RESult][:MEASview<y>]:DUTRatio?	Returns the duty factor (ratio) in the results table.
READ:PULSe[:RESult][:MEASview<y>]:FALL?	Returns the fall time in the results table.
READ:PULSe[:RESult][:MEASview<y>]:FDELta?	Returns the delta frequency in the results table.
READ:PULSe[:RESult][:MEASview<y>]:FRDeviation?	Returns the frequency deviation in the results table.
READ:PULSe[:RESult][:MEASview<y>]:IRAMplitude?	Returns the pulse impulse response amplitude in the pulse table.
READ:PULSe[:RESult][:MEASview<y>]:IRTime?	Returns the impulse response time in the pulse table.
READ:PULSe[:RESult][:MEASview<y>]:MFReqerror?	Returns the maximum frequency error in the results table.
READ:PULSe[:RESult][:MEASview<y>]:MPHerror?	Returns the maximum phase error in the results table.
READ:PULSe[:RESult][:MEASview<y>]:OVEDb?	Returns the overshoot in dB in the pulse table.
READ:PULSe[:RESult][:MEASview<y>]:OVERshoot?	Returns the overshoot in the pulse table.
READ:PULSe[:RESult][:MEASview<y>]:PHDeviation?	Returns the phase deviation in the results table.
READ:PULSe[:RESult][:MEASview<y>]:PPFRequency?	Returns the pulse-pulse carrier frequency in the results table.
READ:PULSe[:RESult][:MEASview<y>]:PPOWer?	Returns the peak power in the results table.
READ:PULSe[:RESult][:MEASview<y>]:PPPHasE?	Returns the pulse-pulse carrier phase in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RINTerval?	Returns the repetition interval in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RIPDb?	Returns the ripple in dB in the pulse table.
READ:PULSe[:RESult][:MEASview<y>]:RIPple?	Returns the ripple in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RISE?	Returns the rise time in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RMSFReqerror?	Returns the RMS frequency error in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RMSPherror?	Returns the RMS phase error in the results table.
READ:PULSe[:RESult][:MEASview<y>]:RRATe?	Returns the repetition rate in the results table.
READ:PULSe[:RESult][:MEASview<y>]:TIME?	Returns the time in the results table.
READ:PULSe[:RESult][:MEASview<y>]:WIDTh?	Returns the pulse width in the results table.
READ:PULSe:STATistics[:MEASview<y>]?	Returns the trace data of the pulse statistics measurement.
READ:PULSe:STATistics[:MEASview<y>]:ATX?	Returns the average transmitted power of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:AVERage?	Returns the average on power of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:DRODb?	Returns the droop in dB of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:DROop?	Returns the droop of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:DUTPct?	Returns the duty factor (%) of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:DUTRatio?	Returns the duty factor (ratio) of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:FALL?	Returns the fall time of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:FDELta?	Returns the delta frequency in the pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:FRDeviation?	Returns the frequency deviation of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:IRAMplitude?	Returns the Pulse Impulse Response amplitude measurement in the pulse statistics.
READ:PULSe:STATistics[:MEASview<y>]:IRTime?	Returns the impulse response time in the pulse statistics.

Table continued...

Command	Description
READ:PULSe:STATistics[:MEASview<y>]:MFRerror?	Returns the maximum frequency error of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:MPHerror?	Returns the maximum phase error of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:OVEdB?	Returns the overshoot measurement result in dB in the pulse statistics
READ:PULSe:STATistics[:MEASview<y>]:OVERshoot?	Returns the overshoot measurement result in the pulse statistics
READ:PULSe:STATistics[:MEASview<y>]:PHDeviation?	Returns the phase deviation of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:PPFRequency?	Returns the pulse-pulse carrier frequency of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:PPOWer?	Returns the peak power of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:PPHase?	Returns the pulse-pulse carrier phase of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RINterval?	Returns the repetition interval of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RIPDb?	Returns the ripple in dB of the statistics
READ:PULSe:STATistics[:MEASview<y>]:RIPple?	Returns the ripple of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RISE?	Returns the rise time of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RMSFreqerror?	Returns the RMS frequency error of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RMSPherror?	Returns the RMS phase error of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:RRATe?	Returns the repetition rate of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:WIDTh?	Returns the pulse width of the statistics.
READ:PULSe:STATistics[:MEASview<y>]:X?	Returns the horizontal values of the trace data of the pulse statistics measurement.
READ:PULSe:STATistics[:MEASview<y>]:XY?	Returns the horizontal and vertical values of the trace data of the pulse statistics measurement.
READ:PULSe:STATistics[:MEASview<y>][:Y]?	Returns the vertical values of the trace data of the pulse statistics measurement.
READ:PULSe:TRACe[:MEASview<y>]:X?	Returns the time values of the pulse trace.
READ:PULSe:TRACe[:MEASview<y>]:XY?	Returns the horizontal (time) and vertical value pairs of the pulse trace.
READ:PULSe:TRACe[:MEASview<y>][:Y]?	Returns the vertical values of the pulse trace.
<b>READ:SEM subgroup</b>	<b>Spectral Emissions Mask measurement</b>
READ:SEM:MHITs:COUnT?	Returns the number of rows in the results table.
READ:SEM:MHITs<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified mask hit signal.
READ:SEM:MHITs<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified mask hit signal.
READ:SEM:MHITs<x>:FREQuency?	Returns the frequency of the specified mask hit signal.
READ:SEM:MHITs<x>:INTeg:ABSolute?	Returns the absolute integration amplitude of the specified mask hit signal.
READ:SEM:MHITs<x>:INTeg:RELative?	Returns the relative integration amplitude of the specified mask hit signal.
READ:SEM:MHITs<x>:MARGin:ABSolute?	Returns the absolute margin for the specified mask hit signal.
READ:SEM:MHITs<x>:MARGin:RELative?	Returns the relative margin for the specified mask hit signal.
READ:SEM:MHITs<x>:OFFSet?	Returns the offset for the specified mask hit signal.

Table continued...

Command	Description
READ:SEM:PASS?	Returns the pass/fail limit test result.
READ:SEM:REF:POWer?	Returns the reference power level.
READ:SEM:SPECtrum:X?	Returns the frequencies of the spectrum trace.
READ:SEM:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
READ:SEM:SPECtrum[:Y]?	Returns the amplitude of the spectrum trace.
<b>READ:SGRam subgroup</b>	<b>Spectrogram measurement</b>
READ:SGRam?	Returns the spectrogram trace data.
<b>READ:SPECtrum subgroup</b>	<b>Spectrum measurement</b>
READ:SPECtrum:TRACe<x>?	Returns the trace data in the Spectrum Analyzer measurement.
<b>READ:SPURious subgroup</b>	<b>Spurious measurement</b>
READ:SPURious:CARRier:POWer?	Returns the carrier power.
READ:SPURious:COUNT?	Returns the number of spurious signals.
READ:SPURious:PASS?	Returns the pass/fail limit test result.
READ:SPURious:SPECtrum:X?	Returns the frequencies of the spectrum trace.
READ:SPURious:SPECtrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace.
READ:SPURious:SPECtrum[:Y]?	Returns the amplitudes of the spectrum trace.
READ:SPURious:SPUR<x>:AMPLitude:ABSolute?	Returns the absolute amplitude of the specified spurious signal.
READ:SPURious:SPUR<x>:AMPLitude:RELative?	Returns the relative amplitude of the specified spurious signal.
READ:SPURious:SPUR<x>:FREQuency:ABSolute?	Returns the absolute frequency of the specified spurious signal.
READ:SPURious:SPUR<x>:FREQuency:RELative?	Returns the relative frequency of the specified spurious signal.
READ:SPURious:SPUR<x>:LIMit:ABSolute?	Returns the absolute amplitude of the limit for a spurious signal.
READ:SPURious:SPUR<x>:LIMit:RELative?	Returns the relative amplitude of the limit for a spurious signal.
READ:SPURious:SPUR<x>:LIMit:VIOLation?	Returns whether the specified spurious signal exceeds the limit or not.
READ:SPURious:SPUR<x>:RANGe?	Returns the frequency range in which the spurious signal occurred.
<b>READ:SQUality subgroup (Option 21 only)</b>	<b>Signal quality measurement</b>
READ:SQUality[:MEASview<y>]:FREQuency:DEViation?	Returns the frequency deviation in the signal quality measurement.
READ:SQUality[:MEASview<y>]:FREQuency:DEViation:TABLE?	Returns the frequency deviation measurement results table.
READ:SQUality[:MEASview<y>]:FREQuency:ERRor?	Returns the frequency error in the signal quality measurement.
READ:SQUality[:MEASview<y>]:GAIN:IMBalance?	Returns the gain imbalance in the signal quality measurement.
READ:SQUality[:MEASview<y>]:ORIGin:OFFSet?	Returns the origin offset in the signal quality measurement.
READ:SQUality[:MEASview<y>]:PEAK:EVM?	Returns the peak EVM (%) in the signal quality measurement.
READ:SQUality[:MEASview<y>]:PEAK:EVM:DB?	Returns the peak EVM (dB) in the signal quality measurement.
READ:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet?	Returns the peak offset EVM (dB) in the signal quality measurement.
READ:SQUality[:MEASview<y>]:PEAK:EVM:LOCation?	Returns the time at which the EVM is peak.
READ:SQUality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet?	Returns the time at which the offset EVM is peak.
READ:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet?	Returns the peak offset EVM (%) in the signal quality measurement.

Table continued...

Command	Description
READ:SQuality[:MEASview<y>]:PEAK:FERRor?	Returns the peak FSK error in the signal quality measurement.
READ:SQuality[:MEASview<y>]:PEAK:MERRor?	Returns the peak magnitude error (%) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:PEAK:MERRor:DB?	Returns the peak magnitude error (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:PEAK:MERRor:LOCation?	Returns the time at which the magnitude error is peak.
READ:SQuality[:MEASview<y>]:PEAK:PERRor?	Returns the peak phase error in the signal quality measurement.
READ:SQuality[:MEASview<y>]:PEAK:PERRor:LOCation?	Returns the time at which the phase error is peak.
READ:SQuality[:MEASview<y>]:QUADrature:ERRor?	Returns the quadrature error in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RHO?	Returns the $\rho$ (waveform quality) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:EVM?	Returns the RMS EVM (%) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:EVM:DB?	Returns the RMS EVM (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:EVM:DB:OFFSet?	Returns the RMS offset EVM (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:EVM:OFFSet?	Returns the RMS offset EVM (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:FERRor?	Returns the RMS FSK error in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:MER:DB?	Returns the RMS MER (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:MERRor?	Returns the RMS magnitude error (%) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:MERRor:DB?	Returns the RMS magnitude error (dB) in the signal quality measurement.
READ:SQuality[:MEASview<y>]:RMS:PERRor?	Returns the RMS phase error in the signal quality measurement.
READ:SQuality[:MEASview<y>]:SYMBOL:LENGth?	Returns the number of analyzed symbols.
READ:SQuality[:MEASview<y>]:SYMBOL:RATE?	Returns the calculated symbol rate in Hz.
READ:SQuality[:MEASview<y>]:SYMBOL:RATE:ERRor?	Returns the value of the symbol rate error in percent.
<b>READ:TDiagram subgroup (Option 21 only)</b>	<b>Trellis diagram</b>
READ:TDiagram[:MEASview<y>]:FERRor?	Returns the frequency error.
READ:TDiagram[:MEASview<y>]:TRACe?	Returns the Trellis diagram trace data.
<b>READ:TOVerview subgroup</b>	<b>Time overview</b>
READ:TOVerview[:MEASview<y>]?	Returns the trace data.
<b>READ:WLAN subgroup</b>	<b>Wireless LAN measurements</b>
READ:WLAN:CONStE:MAGNitude?	Returns the magnitude data for the WLAN Constellation measurement.
READ:WLAN:CONStE:PHASe?	Returns the phase values for the WLAN Constellation measurement.
READ:WLAN:CONStE:TYPE?	Returns the type (context) for the WLAN Constellation measurement.
READ:WLAN:CONStE:VALue?	Returns the value (decoded symbol) for the WLAN Constellation measurement.

Table continued...

Command	Description
READ:WLAN:CRESPonse:MAGNitude?	Returns the magnitude data of the WLAN Channel Response measurement.
READ:WLAN:CRESPonse:PHASe?	Returns the phase data of the WLAN Channel Response measurement.
READ:WLAN:EVM:TRACe<x>?	Returns the trace data for the selected WLAN EVM trace.
READ:WLAN:FLATness:PASS?	Verifies the average power level values of the subcarriers remain within the limits defined for a particular standard.
READ:WLAN:FLATness:RESult?	Returns the summary results of each segment of the WLAN Spectral Flatness display.
READ:WLAN:FLATness:TRACe<x>?	Returns the WLAN Spectral Flatness trace data.
READ:WLAN:MERRor:TRACe<x>?	Returns the trace data for the selected WLAN Magnitude Error trace.
READ:WLAN:PERRor:TRACe<x>?	Returns the trace data for the selected WLAN Phase Error trace.
READ:WLAN:PVTime:BURSt:POWER?	Returns the average power of the burst packet.
READ:WLAN:PVTime:BURSt:WIDTh?	Returns the value of the measured time width of the burst packet from Power-On to Power-Down.
READ:WLAN:PVTime:FALL:TIME?	Returns the width of the falling edge.
READ:WLAN:PVTime:FALL:TIME:START?	Returns the start time of the falling edge.
READ:WLAN:PVTime:FALL:TIME:STOP?	Returns the stop time of the falling edge.
READ:WLAN:PVTime:FTIME?	Returns the width of the falling edge.
READ:WLAN:PVTime:RISE:TIME?	Returns the width of the rising edge.
READ:WLAN:PVTime:RISE:TIME:START?	Returns the start time of the rising edge.
READ:WLAN:PVTime:RISE:TIME:STOP?	Returns the stop time of the rising edge.
READ:WLAN:PVTime:RTIME?	Returns the width of the rising edge.
READ:WLAN:PVTime:TRACe:X?	Returns the horizontal values.
READ:WLAN:PVTime:TRACe:XY?	Returns the horizontal value (time in seconds) and vertical value (power).
READ:WLAN:PVTime:TRACe[Y]?	Returns the vertical values (power).
READ:WLAN:STABLE:VALUe?	Returns the number of bytes in the symbol table.
READ:WLAN:SUMMary:BURSt:INDex?	Returns the index of the analyzed packet.
READ:WLAN:SUMMary:BURSt:POWER?	Returns the average power of all symbols in the packet.
READ:WLAN:SUMMary:BURSt:POWER:CFACTOR?	Returns the peak-to-average burst power factor.
READ:WLAN:SUMMary:CERRor?	Returns the symbol clock error.
READ:WLAN:SUMMary:CPE?	Returns the RMS magnitude error of the pilots.
READ:WLAN:SUMMary:EVM:PEAK:ALL?	Returns the peak EVM value for all subcarriers or all chips.
READ:WLAN:SUMMary:EVM:PEAK:CHIP:ALL?	Returns the chip at which the reported peak EVM value occurred.
READ:WLAN:SUMMary:EVM:PEAK:CHIP:ONEKchips?	Returns the chip at which the reported peak EVM value over the first 1000 chips occurred.
READ:WLAN:SUMMary:EVM:PEAK:DATA?	Returns the peak EVM value for the data subcarriers.

Table continued...

Command	Description
READ:WLAN:SUMMARY:EVM:PEAK:ONEKchips?	Returns the peak EVM value measured over the first 1000 chips.
READ:WLAN:SUMMARY:EVM:PEAK:PILots?	Returns the peak EVM value for the pilots in the WLAN measurement.
READ:WLAN:SUMMARY:EVM:PEAK:SCARrier:ALL?	Returns the subcarrier on which the ALL peak EVM value occurred.
READ:WLAN:SUMMARY:EVM:PEAK:SCARrier:DATA?	Returns the subcarrier on which the DATA peak EVM value occurred.
READ:WLAN:SUMMARY:EVM:PEAK:SCARrier:PILots?	Returns the subcarrier on which the PILOT peak EVM value occurred.
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:ALL?	Returns the symbol on which the reported EVM value occurred.
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:DATA?	Returns the symbol on which the DATA peak EVM value occurred.
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:PILots?	Returns the symbol on which the PILOT peak EVM value occurred.
READ:WLAN:SUMMARY:EVM:RMS:ALL?	Returns the RMS EVM data for all subcarriers.
READ:WLAN:SUMMARY:EVM:RMS:AVERAGE?	Returns the average RMS EVM value.
READ:WLAN:SUMMARY:EVM:RMS:DATA?	Returns the RMS EVM data for the data subcarriers.
READ:WLAN:SUMMARY:EVM:RMS:MAXimum?	Returns the maximum RMS EVM value.
READ:WLAN:SUMMARY:EVM:RMS:ONEKchips?	Returns the RMS EVM data for the 1k chips in the WLAN measurement measured over the first 1000 chips.
READ:WLAN:SUMMARY:EVM:RMS:PILots?	Returns the RMS EVM data for the pilot subcarriers.
READ:WLAN:SUMMARY:FERRor?	Returns the difference between the measured carrier frequency of the signal and the measurement frequency setting.
READ:WLAN:SUMMARY:HEADer?	Returns the received packet header information.
READ:WLAN:SUMMARY:HTSig?	Returns the decimal values of the received packet HT-SIGNAL
READ:WLAN:SUMMARY:IQOffset?	Returns the average magnitude of the DC subcarrier level relative to total signal power.
READ:WLAN:SUMMARY:LSIG?	Returns the decimal values of the received packet (legacy) SIGNAL symbols fields.
READ:WLAN:SUMMARY:PACKet?	Returns the type and number of symbols in the Preamble and Data portions of the packet.
READ:WLAN:SUMMARY:PACKet:DMODulation?	Returns the modulation used in the Data symbols.
READ:WLAN:SUMMARY:PACKet:FORMat?	Returns the packet format setting.
READ:WLAN:SUMMARY:PACKet:GUARd?	Returns the packet guard interval used by the DATA symbols.
READ:WLAN:SUMMARY:SIG?	Returns the decimal values of the received packet SIGNAL symbols fields.
READ:WLAN:SUMMARY:VHTSig:A?	Returns the decimal values of the received packet VHT-SIGNAL(A) symbols fields.
READ:WLAN:SUMMARY:VHTSig:B?	Returns the decimal values of the received packet VHT-SIGNAL(B) symbols fields.
<b>READ:LTE subgroup</b>	
READ:LTE:ACLR:MHITs:COUNt?	Returns the number of rows in the results table.
READ:LTE:ACLR:MHITs<x>:BANDwidth?	Returns the bandwidth value of the row (channel).
Table continued...	



Command	Description
READ:LTE:ACLR:MHITs<x>:CHANnel:NAME?	Returns the channel name of the specified index in the results table.
READ:LTE:ACLR:MHITs<x>:FREQuency:OFFSet?	Returns the frequency offset of the channel.
READ:LTE:ACLR:MHITs<x>:INTeg:LOWEr:ABSolute?	Returns the lower absolute value of the channel.
READ:LTE:ACLR:MHITs<x>:INTeg:LOWEr:RELative?	Returns the lower relative value of the channel.
READ:LTE:ACLR:MHITs<x>:INTeg:UPPEr:ABSolute?	Returns the upper absolute value of the channel.
READ:LTE:ACLR:REFerence:POWEr?	Returns the frequency and amplitude pairs of the spectrum trace.
READ:LTE:ACLR:RESUlts:STATus?	Returns the pass or fail status.
READ:LTE:ACLR:SPEctrum:X?	Returns the frequency of the spectrum trace.
READ:LTE:ACLR:SPEctrum:XY?	Returns the frequency and amplitude pairs of the spectrum trace
READ:LTE:ACLR:SPEctrum:Y?	Returns the amplitude of the spectrum trace.
READ:LTE:CHSPectrum:OBW?	Returns the Occupied Bandwidth (Hz).
READ:LTE:CHSPectrum:POWEr:CHANnel?	Returns the Channel Power value.
READ:LTE:CHSPectrum:SPEctrum?	Returns spectrum trace data.
READ:LTE:CONStE:CELL:ID?	Queries the Cell ID value.
READ:LTE:CONStE:FREQuency:ERRor?	Queries Frequency Error value in Hz.
READ:LTE:CONStE:GROUp:ID?	Queries the Group ID value.
READ:LTE:CONStE:SECTor:ID?	Queries the Sector ID value.
READ:LTE:PVTime:OFFSlot:POWEr?	Returns the TOff power measurement value in dBm/MHz.
READ:LTE:PVTime:RESUlts:STATus	Returns the pass or fail status.
READ:LTE:PVTime:TRACe:X?	Returns the horizontal values (time in seconds).
READ:LTE:PVTime:TRACe:XY?	Returns the horizontal value (time in seconds) and vertical value (power).
READ:LTE:PVTime:TRACe:Y?	Returns the vertical values (power).

## Sense commands

Use the SENSE commands to set up detailed measurement conditions. See the Index for commands not listed in the following table.

The Record and Playback command subgroup may not apply to your instrument. Details can be found in the command descriptions.

**Table 23: Sense commands**

Command	Description
<b>[SENSe] basic command subgroup</b>	<b>General analysis parameter control</b>
[SENSe]:ACQuisition:{BANDwidth BWIDth}	Sets or queries the acquisition bandwidth.
<b>[SENSe]:ACQuisition:RECOrd command subgroup</b>	<b>Record and Playback function for USB RSA products</b>
[SENSe]:ACQuisition:RECOrd:FILE:LENGth	Sets or queries how much data (time in milliseconds, seconds, or minutes) is recorded when recording starts.

Table continued...

Command	Description
[SENSe]:ACQuisition:RECOrd:FILE:LENGth:ENABle	Sets or queries the file length conditions to determine how much data is recorded.
[SENSe]:ACQuisition:RECOrd:FILE:LENGth:UNITs	Sets or queries the units used for the file length setting.
[SENSe]:ACQuisition:RECOrd:FILE:LOCation	Sets the file location of the saved data records.
[SENSe]:ACQuisition:RECOrd:FILE:MAXimum	Sets or queries how many times a record is created, of the specified File Length setting.
[SENSe]:ACQuisition:RECOrd:FILE:NAME:BASE	Sets or queries the base file name that is used for all data recordings.
[SENSe]:ACQuisition:RECOrd:FILE:STRucture	Sets or queries the file structure to use for the recorded data files.
[SENSe]:ACQuisition:RECOrd:STOP	Starts the data recording.
[SENSe]:ACQuisition:RECOrd:TYPE	Sets or queries when the recording starts streaming data to the specified file.
[SENSe]:ANALysis:BANDwidth:FILTER:MODE	Sets or queries measurement bandwidth filter mode.
<b>[SENSe]:AUDIO:DEMod command subgroup</b>	<b>Record and Playback function for USB RSA products</b>
[SENSe]:AUDIO:DEMod:MUTE	Sets or queries the mute setting for the audio demodulation.
[SENSe]:AUDIO:DEMod[:STATe]	Sets or queries the audio demodulation state.
[SENSe]:AUDIO:DEMod:STReam:ACTivate	Sets or queries the Stream to File function.
[SENSe]:AUDIO:DEMod:STReam[:FILE]	Sets the file name and file location of the audio file when streaming to disc.
[SENSe]:AUDIO:DEMod:TYPE	Sets or queries both the audio signal demodulation type and bandwidth.
[SENSe]:AUDIO:DEMod:VOLume	Sets or queries the volume of the demodulated audio signal when played out through the PC speakers.
<b>[SENSe] TXGain command subgroup</b>	<b>Transmission Gain measurements</b>
SENSe:TXGain:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth.
SENSe:TXGain:FREQuency:CENTer	Sets or queries the frequency at the center of the selected Span.
SENSe:TXGain:FREQuency:POINTs	Sets or queries the number of frequency steps in the measurement range.
SENSe:TXGain:FREQuency:SPAN	Sets or queries the difference between the start and stop frequencies (span).
SENSe:TXGain:FREQuency:START	Sets or queries the lowest frequency in the span.
SENSe:TXGain:FREQuency:STEP	Sets or queries the frequency increment.
SENSe:TXGain:FREQuency:STOP	Sets or queries the highest frequency in the span.
<b>[SENSe] Bluetooth command subgroup</b>	<b>Bluetooth measurements</b>
[SENSe]:BLUETOOTH:ANALysis:LENGth	Sets or returns the analysis length.
[SENSe]:BLUETOOTH:ANALysis:LENGth:ACTual?	Sets the actual analysis length.
[SENSe]:BLUETOOTH:ANALysis:LENGth:AUTO	Sets or queries the length mode.
[SENSe]:BLUETOOTH:ANALysis:MEAS[:BANDwidth BWIDth]	Sets or returns the measurement bandwidth.
[SENSe]:BLUETOOTH:ANALysis:MEAS:SOURce	Sets or returns the analysis measurement source.

Table continued...

Command	Description
[SENSe]:BLUEtooth:ANALysis:OFFSet	Sets or returns the symbol analysis offset.
[SENSe]:BLUEtooth:ANALysis:OFFSet:AUTO	Determines whether to set the carrier frequency error automatically or manually.
[SENSe]:BLUEtooth:ANALysis:TIME:UNITS	Sets the time units.
[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):MODE	Sets or returns the measurement bandwidth mode.
[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):TINterval	Sets or returns the measurement bandwidth frequency span.
[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):TINterval:AUTO	Sets or returns the bandwidth to Auto or Manual.
[SENSe]:BLUEtooth:CONStellation:PREFs:GRATICule:SHOW	Show or hides the graticule state.
[SENSe]:BLUEtooth:CONStellation:PREFs:MARKERS:SHOW	Show or hides the marker readout.
[SENSe]:BLUEtooth:CONStellation:PREFs:RADix	Sets or returns the radix of the marker readout.
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:CONTent	Sets or returns the appearance of the specified trace (Trace 1 or Trace 2).
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:FREEze	Sets or freezes the updates of the specified trace (Trace 1 or Trace 2).
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:POINtsPerSymbol	Sets or returns the number of points per symbol (how many points to use between symbols) for the specified trace (Trace 1 or Trace 2).
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:SHOW	Shows or hides the specified trace (Trace 1 or Trace 2).
[SENSe]:BLUEtooth:DELTA:AVERage:FONE	Sets or returns the average frequency for the low deviation pattern.
[SENSe]:BLUEtooth:DELTA:AVERage:FTWO	Sets or returns the average frequency for the high deviation pattern.
[SENSe]:BLUEtooth:FDVTime:TRACe:OCTET?	Returns the selected octet number on the Frequency Deviation vs. Time display.
[SENSe]:BLUEtooth:FDVTime:TRACe:OCTET:TOTAL: COUNT?	Returns the total octet value to be selected on the Frequency Deviation vs. Time display.
[SENSe]:BLUEtooth:FDVTime:TRACe:VIEW:MODE	Sets or queries view mode in the Frequency Deviation vs. Time display.
[SENSe]:BLUEtooth:FILTer:ALPHA	Sets or returns the alpha filter factor ( $\alpha$ /BT).
[SENSe]:BLUEtooth:FILTer:MEASurement	Sets or returns the measurement filter setting.
[SENSe]:BLUEtooth:FILTer:REFerence	Sets or returns the reference filter setting.
[SENSe]:BLUEtooth:FREQUency:DEVIation	ets or returns the frequency deviation.
[SENSe]:BLUEtooth:FREQUency:DEVIation:AUTO	Determines whether to detect the frequency deviation automatically or manually.
[SENSe]:BLUEtooth:FREQUency:ERRor	Sets or returns the frequency error in the Bluetooth analysis parameters (when Auto is off).
[SENSe]:BLUEtooth:FREQUency:ERRor:AUTO	Determines whether the frequency error is set to Auto or Manual.
[SENSe]:BLUEtooth:INBEmissions:POWER:LIMIt<x>	Sets or queries the limits of the Bluetooth InBand Emission settings.
[SENSe]:BLUEtooth:POWer:CLASs	Sets or queries the power class of the Bluetooth standard.
[SENSe]:BLUEtooth:STANdard	Sets or queries the Bluetooth standard.
[SENSe]:BLUEtooth:SUMMery:LIMIt:BDR:DELTA:FONE:AVERage:MAX	Sets or returns the maximum limit of the Basic Rate $\Delta F1$ average.
Table continued...	

Command	Description
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MIN	Sets or returns the minimum limit of the Basic Rate $\Delta F1$ average.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:PERcent	Sets or returns the limit of the Basic Rate $\Delta F2$ percentage.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:RESult	Sets or returns the limit of the Basic Rate $\Delta F2$ result.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:DRIFt:FNFZero	Sets or returns the limit of the Basic Rate maximum drift $f_n - f_0$ .
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQUency:OFFSet	Sets or queries the limit of the Basic Rate maximum frequency offset.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQUency:OFFSet:PREAmBDR	Sets or queries the limit of the Basic Rate maximum frequency offset value from the preamble region.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MAX	Sets or queries the limit of the Basic Rate Power Class 1 maximum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MIN	Sets or queries the limit of the Basic Rate Power Class 1 minimum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MAX	Sets or queries the limit of the Basic Rate Power Class 1 maximum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MIN	Sets or queries the limit of the Basic Rate Power Class 1 minimum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:THREe:AVERAge:MAX	Sets or queries the limit of the Basic Rate Power Class 3 maximum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:THREe:AVERAge:MIN	Sets or queries the limit of the Basic Rate Power Class 3 minimum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MAX	Sets or queries the limit of the Basic Rate Power Class 3 maximum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MIN	Sets or queries the limit of the Basic Rate Power Class 3 minimum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MAX	Sets or queries the limit of the Basic Rate Power Class 2 maximum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MIN	Sets or queries the limit of the Basic Rate Power Class 2 minimum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MAX	Sets or queries the limit of the Basic Rate Power Class 2 maximum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MIN	Sets or queries the limit of the Basic Rate Power Class 2 minimum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MAX	Sets or returns the maximum limit of the Low Energy $\Delta F1$ average.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MIN	Sets or returns the minimum limit of the Low Energy $\Delta F1$ average.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FTWO:PERcent	Sets or returns the limit of the Low Energy $\Delta F2$ percentage.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FTWO:RESult	Sets or returns the limit of the Low Energy $\Delta F2$ result.
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:DRIFt:FNFZero	Sets or returns the limit of the Low Energy maximum drift $f_n - f_0$ .
[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQUency:OFFSet	Sets or returns the limit of the Low Energy maximum frequency offset.

Table continued...

Command	Description
[SENSe]:BLUEtooth:SUMMary:LIMit:BLE:MAX:FREQUency:OFFSet:PREAmble	Sets or returns the limit of the Low Energy maximum frequency offset value from the preamble region.
[SENSe]:BLUEtooth:SUMMary:LIMit:BLE:OPower:AVERage:MAX	Sets or returns the limit of the Low Energy maximum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMit:BLE:OPower:AVERage:MIN	Sets or returns the limit of the Low Energy minimum average output power value.
[SENSe]:BLUEtooth:SUMMary:LIMit:BLE:OPower:PEAK:MAX	Sets or returns the limit of the Low Energy maximum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMit:BLE:OPower:PEAK:MIN	Sets or returns the limit of the Low Energy minimum peak output power value.
[SENSe]:BLUEtooth:SUMMary:LIMit:MAX:DRIFT:F1FZero	Sets or returns the limit of the maximum $f_n-f_{n0}$ drift in the packet.
[SENSe]:BLUEtooth:SUMMary:LIMit:MAX:DRIFT:FNF5	Sets or returns the limit of the maximum $f_n-f_{n5}$ drift in the packet.
[SENSe]:BLUEtooth:SYMBol:POINts	Sets or returns the number of points per symbol (how many points to use between symbols).
[SENSe]:BLUEtooth:TEST:PATtern	Sets or returns the test pattern type in the Bluetooth displays.
[SENSe]:BLUEtooth:TEST:PATtern:AUTO:DETEct	Sets or returns the auto detect setting for the Bluetooth displays.
[SENSe]:BLUEtooth:TIME:ZERO:REFERENCE	Sets or returns the Time Zero Reference setting in the Bluetooth displays.
[SENSe]:BLUEtooth:TRACe:TYPE	Sets or returns the trace type.
<b>SENSe:EDDiagram subgroup</b>	<b>Eye diagram</b>
[SENSe]:EDDiagram[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source of Eye Diagram for a selected Measurement view number.
<b>SENSe:EVM subgroup</b>	<b>EVM measurement</b>
[SENSe]:DEMOd:EVM[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source of EVM versus Time for a selected Measurement view number.
<b>SENSe:FDVTime subgroup</b>	<b>FDVTime measurement</b>
[SENSe]:FDVTime[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source of Frequency deviation versus Time for a selected Measurement view number
<b>[SENSe]:MERRor subgroup</b>	<b>MERRor measurement</b>
[SENSe]:MERRor[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source of Magnitude error versus Time for a selected Measurement view number
<b>[SENSe]: NOISe subgroup</b>	<b>Noise Figure and Gain measurements</b>
[SENSe]:NOISe:AVERage:COUNT	Sets or queries the acquisition counts over which the noise average is computed.
[SENSe]:NOISe:BANDwidth	Sets or queries the resolution bandwidth (RBW).
[SENSe]:NOISe:CONTRol:SETTling:OFF[:TIME]	Sets or queries the analyzer off settle time.
[SENSe]:NOISe:CONTRol:SETTling:ON[:TIME]	Sets or queries the analyzer on settle time.
[SENSe]:NOISe:CORREction:ENR:CONSTant	Sets and queries the excess noise ratio (ENR) constant.
[SENSe]:NOISe:CORREction:ENR:MODE	Sets and queries the Noise Source mode for the noise displays.
[SENSe]:NOISe:CORREction:ENR:TABLE:DATA?	Queries the Noise Source model, serial number, and table values for frequency and ENR for the noise displays.

Table continued...

Command	Description
[SENSe]:NOISe:CORRection:ENR:TABLE:ID	Sets or queries the Noise Source Model for the noise displays.
[SENSe]:NOISe:CORRection:ENR:TABLE:NEW	Sets the Noise Source Model for the noise displays.
[SENSe]:NOISe:CORRection:ENR:TABLE:SERial	Sets or queries the Noise Source serial number for the noise displays.
[SENSe]:NOISe:CORRection:TCOLd:ADUT:UREFERENCE	Sets or queries the analyzer to use the reference temperature (K) for the loss at the DUT output in the noise displays.
[SENSe]:NOISe:CORRection:TCOLd:ADUT[:VALue]	Sets or queries the temperature value (K) for the loss at the DUT output in the noise displays.
[SENSe]:NOISe:CORRection:TCOLd:BDUT:UREFERENCE	Sets or queries the analyzer to use the reference temperature (K) for the loss at the DUT input in the noise displays.
[SENSe]:NOISe:CORRection:TCOLd:BDUT[:VALue]	Sets or queries the temperature value (K) for the loss at the DUT input in the noise displays.
[SENSe]:NOISe:CORRection:TCOLd[:REFERENCE]	Sets or queries the reference temperature value (K) in the display.
[SENSe]:NOISe:FIGure:POINTs:COUNT	Sets the number of sample points for the display.
[SENSe]:NOISe:FREQuency:CENTer	Sets or queries the center frequency value (Hz).
[SENSe]:NOISe:FREQuency:MODE	Sets or queries the frequency mode in the display.
[SENSe]:NOISe:FREQuency:SPAN	Sets or queries the frequency Span value (Hz).
[SENSe]:NOISe:FREQuency:START	Sets or queries the Start Frequency value.
[SENSe]:NOISe:FREQuency:STEPS	Sets or queries the frequency Steps value.
[SENSe]:NOISe:FREQuency:STOP	Sets or queries the Stop Frequency value.
[SENSe]:NOISe:FREQuency:TABLE:DATA?	Queries the data in the Frequency Table.
[SENSe]:NOISe:FREQuency:TABLE:NEW	Creates a new Frequency Table with the specified values for the noise display.
[SENSe]:OBWidth:BOBW:XDBM:MEASurement:DIREction	Sets or queries the measurement direction.
[SENSe]:OBWidth:BOBW:XDBM:RANGe	Sets or queries the measurement range.
<b>[SENSe]:P25 subgroup</b>	<b>APCO P25 measurements</b>
[SENSe]:P25:ANALysis:OFFSet	Sets or queries the offset value.
[SENSe]:P25:ANALysis:OFFSet:AUTO	Sets or queries if the offset is set to Auto or if Auto is off.
[SENSe]:P25([:BANDwidth]):BWIDth:MODE	Sets or queries the Measurement BW (bandwidth) (frequency span) mode as Auto, Manual, or Link to Span.
[SENSe]:P25([:BANDwidth]):BWIDth:TINTerval	Sets or queries the Measurement BW (bandwidth) (frequency span).
[SENSe]:P25([:BANDwidth]):BWIDth:TINTerval:AUTO	Sets or queries the Measurement BW (bandwidth) (frequency span) mode as Auto or Manual.
[SENSe]:P25:FILTer:MEASurement	Sets or queries the Measurement Filter setting.
[SENSe]:P25:FILTer:REFERENCE	Sets or queries the Reference Filter setting.
[SENSe]:P25:FREQuency:ERRor	Sets or queries the Frequency Offset (when Auto is off).
[SENSe]:P25:FREQuency:ERRor:AUTO	Sets or queries if the Frequency Error is set to Auto or if Auto is off.
[SENSe]:P25:MODUlation:CHANnel:([:BANDwidth]): BWIDth)?	Queries the actual P25 Preset channel bandwidth.

Table continued...

Command	Description
[SENSe]:P25:MODUlation:STANdard	Sets or queries the modulation standard.
SENSe:P25:MODUlation:TYPE	Sets or queries the modulation type.
[SENSe]:P25:SRATe	Sets or queries the symbol rate value for measurements.
[SENSe]:P25:SUMMary:CLEar:RESults	Clears the results from the P25 Summary display.
[SENSe]:P25:SUMMary:LIMIt:FA	Sets or queries the On/Off state of limit comparison and the upper limit value for Operating Frequency Accuracy.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:AVERAge:TE RRor:ONE	Sets or queries the On/Off state of limit comparison and the upper limit value for Average t_error_1.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:AVERAge:TE RRor:ZERO	Sets or queries the On/Off state of limit comparison and the upper limit value for Average t_error_0.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:LOGIc:CHAN nel:OFF:SLOT	Sets or queries the On/Off state of limit comparison and the upper limit value for HCPM Transmitter Logic channel offslot.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:LOGIc:CHAN nel:PEAK:ACPR	Sets or queries the On/Off state of limit comparison and the upper limit value for HCPM Transmitter Logic channel Peak ACPR.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PEAK:NEGAti ve:LO	Sets or queries the On/Off state of limit comparison and the lower limit value for Frequency Deviation, negative peak.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PEAK:NEGAti ve:UP	Sets or queries the On/Off state of limit comparison and the upper limit value for Frequency Deviation, negative peak.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PEAK:POSIti ve:LO	Sets or queries the On/Off state of limit comparison and the lower limit value for Frequency Deviation, positive peak.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PEAK:POSIti ve:UP	Sets or queries for the On/Off state of limit comparison and the upper limit value for Frequency Deviation, positive peak.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PHASe[1/2]: F[1/2/3/4]:LO	Sets or queries the On/Off state of limit comparison and the lower limit value for Frequency Deviation, f1/f2/f3/f4 for Phase1 or Phase2.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PHASe[1/2]: F[1/2/3/4]:UP	Sets or queries the On/Off state of limit comparison and the upper limit value for Frequency Deviation, f1/f2/f3/f4 for Phase1 or Phase2.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PMAX:OFF	Sets or queries the On/Off state of limit comparison and the upper limit value for Pmax-off.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PMAX:ON	Sets or queries the On/Off state of limit comparison and the upper limit value for Pmax-on.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PMAX:SS	Sets or queries the On/Off state of limit comparison and the upper limit value for Pss-max.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:PMIN:SS	Sets or queries the On/Off state of limit comparison and the upper limit value for Pss-min.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:TRANsmitter: AVERAge:ATTK:ENCOder	Sets or queries the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack encoder.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:TRANsmitter: AVERAge:ATTK:ENCOder:BI	Sets or queries the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack encoder, busy/idle.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:TRANsmitter: AVERAge:ATTK:POWER	Sets or queries the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack power.
Table continued...	

Command	Description
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:TRANsmitter:AVERAge:ATTK:POWEr:BI	Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack power, busy/idle.
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVlation:TRANsmitter:AVERAge:THRUput:DLAY	Sets or queries the On/Off state of limit comparison and the upper limit value for Transmitter Average throughput delay.
[SENSe]:P25:SUMMary:LIMIt:MF	Sets or queries the On/Off state of limit comparison and the upper limit value for Modulation Fidelity.
[SENSe]:P25:SUMMary:LIMIt:RF:OUTPower	Sets or queries the On/Off state of limit comparison and the upper limit value for RF Output Power.
[SENSe]:P25:SUMMary:LIMIt:SRA	Sets or queries the On/Off state of limit comparison and the upper limit value for Symbol Rate Accuracy.
[SENSe]:P25:SYMBols:POINts	Sets or queries the Points/Symbols to display.
[SENSe]:P25:TIME:UNITs	Sets or queries the fundamental unit of time for the P25 measurements.
<b>[sense]:acquisition subgroup</b>	<b>Acquisition measurements</b>
[SENSe]:SVPC:ACQuisition:FFRame:ACTual?	Queries the actual number of Fast Frames. (When using SignalVu-PC.)
[SENSe]:SVPC:ACQuisition:FFRame:LIMit	Sets or queries the limit number of Fast Frames. (When using SignalVu-PC.)
[SENSe]:SVPC:ACQuisition:FFRame:STATe	Determines whether to enable or disable the Fast Frame. (When using SignalVu-PC.)
[SENSe]:ACQuisition:FSAVe:ENABle	Set or queries the FastSave acquisition mode.
[SENSe]:ACQuisition:FSAVe:FILEs:MAXimum	Set or queries the FastSave maximum number of files saved per run.
[SENSe]:ACQuisition:FSAVe:FORMat	Set or queries the FastSave file format.
[SENSe]:ACQuisition:FSAVe:LOCation	Set or queries the FastSave file save location (a directory).
[SENSe]:ACQuisition:FSAVe:NAME:BASE	Set or queries the FastSave Base file name.
[SENSe]:ACQuisition:MEMory:AVAIlable:SAMPles?	Returns the amount of acquisition memory available in the instrument.
[SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?	Returns the acquisition memory capacity.
[SENSe]:ACQuisition:MEMory:USED[:PERCent]?	Returns the percentage of the capacity used.
[SENSe]:ACQuisition:MODE	Sets or queries the acquisition mode.
[SENSe]:ACQuisition:OPTimization	Sets or queries the method of the global gain and input bandwidth optimization.
[SENSe]:ACQuisition:SAMPles	Sets or queries the acquisition samples.
[SENSe]:ACQuisition:SECOnds	Sets or queries the acquisition length.
[SENSe]:ANALysis:ADVanced:DITHer	Determines whether to enable or disable dithering.
[SENSe]:ANALysis:ADVanced:DITHer:HWAre: STATus?	Returns the dithering hardware status.
[SENSe]:ANALysis:LENGth[:SOURce<z>]:ACTual	Sets or queries the analysis length. Programming a specified length sets [SENSe]:ANALysis:AUTO OFF.
[SENSe]:ANALysis:LENGth:ACTual?	Queries the actual analysis length.

Table continued...



Command	Description
[SENSe]:ANALysis:LENGth[:SOURce<z>]:AUTO	Determines whether to set the analysis length automatically or manually.
[SENSe]:ANALysis:REFeRence[:SOURce<z>]	Sets or queries the analysis time reference.
[SENSe]:ANALysis:STARt[:SOURce<z>]	Sets or queries the analysis offset time. Programming a specified offset time sets [SENSe]:ANALysis:STARt:AUTO OFF.
[SENSe]:ANALysis:STARt[:SOURce<z>]:AUTO	Determines whether to set the analysis offset automatically or manually.
[SENSe]:ANSPectrum:STARt:AUTO:METhod	Sets or queries the method used for computing auto analysis and spectrum offsets.
[SENSe]:MEASurement:FREQUency[:SOURce<z>]	Sets or queries the measurement frequency.
[SENSe]:MEASurement:FREQUency[:SOURce<z>]:CENTer:LOCK	Sets or queries locking the measurement center frequency.
[SENSe]:POWER:UNITs	Sets or queries the fundamental unit of power.
[SENSe]:ROSCillator:SOURce	Sets or queries the reference oscillator source.
[SENSe]:SIGNAlpath:PRiority	Sets or queries the signal path priority for the measurements (for RSA5000 series only).
[SENSe]:USETtings	Updates the analyzer settings.
<b>[SENSe]:ACPower subgroup</b>	<b>Channel power and ACPR measurement</b>
[SENSe]:ACPower:AVERAge	Sets or queries how to average waveform.
[SENSe]:ACPower:AVERAge:COUNT	Sets or queries the number of traces for averaging.
[SENSe]:ACPower:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:ACPower:{BANDwidth BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:ACPower:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the RBW automatically.
[SENSe]:ACPower:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:ACPower:{BANDwidth BWIDth}:VIDeo:STATe	Sets or queries whether to enable the video bandwidth (VBW).
[SENSe]:ACPower:CHANnel:{BANDwidth BWIDth}	Sets or queries the channel bandwidth.
[SENSe]:ACPower:CHANnel:FILTer	Sets or queries the adjacent channel filter.
[SENSe]:ACPower:CHANnel:PAIRs	Sets or queries the number of adjacent channel pairs.
[SENSe]:ACPower:CHANnel:SPACing	Sets or queries the channel-to-channel spacing.
[SENSe]:ACPower:CHIPrate	Sets or queries the chip rate.
[SENSe]:ACPower:CLEar:RESults	Restarts the average trace.
[SENSe]:ACPower:FREQUency	Sets or queries the center frequency.
[SENSe]:ACPower:FREQUency:STEP	Sets or queries the frequency step size.
[SENSe]:ACPower:FREQUency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:ACPower:NFLoor:STATe	Sets or queries whether to enable the correction for noise floor.
[SENSe]:ACPower:POINTs:COUNT	Sets or queries the number of trace points.
[SENSe]:ACPower:RRCRolloff	Sets or queries the filter parameter for the Root Raised Cosine filter.
[SENSe]:ACPower:SYMBrate	Sets or queries the symbol rate.
<b>[SENSe]:{AM FM PM} subgroup (Option 21 only)</b>	<b>AM/FM/PM measurement</b>

Table continued...

Command	Description
[SENSe]:{AM FM PM}:{BANDwidth BWIDth}:MEASurement	Sets measurement bandwidth for the AM, FM, or PM demodulation to analyze.
[SENSe]:{AM FM PM}:CLEar:RESults	Clears measurement results.
[SENSe]:{AM FM PM}:{MTPoints MAXTracepoints}	Sets or queries the maximum trace points.
[SENSe]:AM:DETECT:AMPLitude	Sets or queries the carrier amplitude detection method.
[SENSe]:{FM PM}:BURSt:THReshold	Sets or queries the threshold level to detect bursts.
[SENSe]:{FM PM}:FREQuency:OFFSet	Sets or queries the carrier frequency offset in the FM/PM measurement.
[SENSe]:{FM PM}:FREQuency:OFFSet:MARKer	Sets the frequency offset from selected marker or delta marker location.
[SENSe]:{FM PM}:FREQuency:SEARch:AUTO	Sets or queries whether to detect the carrier frequency automatically.
[SENSe]:PM:PHASe:OFFSet	Sets the phase offset from the selected marker location.
[SENSe]:PM:PHASe:OFFSet:MARKer	Sets the carrier frequency offset from the selected marker.
[SENSe]:PM:PHASe:SEARch:AUTO	Sets or queries whether to detect the phase automatically or manually.
<b>{SENSe}:AUDio subgroup</b>	<b>Audio measurement</b>
[SENSe]:AUDio:AVERAge:COUNT	Sets or queries the number of harmonics to average in the audio measurement.
[SENSe]:AUDio:AVERAge:ENABLE	Enables or disables averaging of harmonics in the audio measurement.
[SENSe]:AUDio:BANDwidth BWIDth	Sets or queries the audio measurement bandwidth.
[SENSe]:AUDio:BANDwidth BWIDth:RESolution	Sets or queries the resolution bandwidth for the audio measurement.
[SENSe]:AUDio:BANDwidth BWIDth:RESolution:AUTO	Sets or queries the resolution bandwidth for the audio measurement.
[SENSe]:AUDio:BANDwidth BWIDth:RESolution:SHAPE	Sets or queries the shape of the resolution bandwidth for the audio measurement.
[SENSe]:AUDio:CARRier:OFFSet	Sets or queries the carrier frequency offset.
[SENSe]:AUDio:CARRier:OFFSet:AUTO	Determines whether to set the carrier frequency error automatically or manually.
[SENSe]:AUDio:FILTer:DEEMphasis	Sets or queries the de-emphasis filter type.
[SENSe]:AUDio:FILTer:DEEMphasis:ENABLE	Enables or disables the de-emphasis filter for the audio measurement.
[SENSe]:AUDio:FILTer:DEEMphasis:USER	Sets or queries a user-defined de-emphasis audio filter.
[SENSe]:AUDio:FILTer:HPF	Sets or queries the high-pass filter type.
[SENSe]:AUDio:FILTer:HPF:ENABLE	Enables or disables the high-pass audio filter for the audio measurement.
[SENSe]:AUDio:FILTer:HPF:USER	Sets or queries a user-defined high-pass audio filter.
[SENSe]:AUDio:FILTer:DEEMphasis:USER	Sets or queries a user-defined high-pass audio filter.
[SENSe]:AUDio:FILTer:LPF	Sets or queries the low-pass filter type.

Table continued...

Command	Description
[SENSe]:AUDio:FILTer:LPF:ENABLE	Enables or disables the low-pass filter for the audio measurement.
[SENSe]:AUDio:FILTer:LPF:USER	Sets or queries a user-defined low-pass audio filter.
[SENSe]:AUDio:FILTer:MODE:PREDEfined	Selects a predefined audio filter or a filter stored on the instrument file.
[SENSe]:AUDio:FILTer:SOURce?	Queries the measurement filter from the user-defined source file.
[SENSe]:AUDio:FILTer:STANdard	Sets or queries the Standard audio filter.
[SENSe]:AUDio:FILTer:STANdard:ENABLE	Enables or disables a Standard filter for the audio measurement.
[SENSe]:AUDio:FREQuency	Sets or queries the reference audio frequency.
[SENSe]:AUDio:FREQuency:AUTO	Determines whether to set the reference audio frequency automatically or manually.
[SENSe]:AUDio:HNOise:ENABLE	Enables or disables the Hum and Noise information in the Audio Summary display.
[SENSe]:AUDio:HNOise:REFerence	Captures the current Hum and Noise information in the Audio Summary display.
[SENSe]:AUDio:SIGnal:TYPE	Sets or queries the audio signal type.
[SENSe]:AUDio:SPECtrum:CLEar:RESults	Clears the results of the audio spectrum measurement.
[SENSe]:AUDio:SPECtrum:POINts:COUNT	Sets or queries the number of trace points acquired for the audio spectrum display.
[SENSe]:AUDio:UNITs	Sets or queries the units of the audio spectrum measurement.
<b>[SENSe]:AVTime subgroup</b>	<b>Amplitude versus Time measurement</b>
[SENSe]:AVTime [:MEASview<y>]:{BANDwidth BWIDth}	Sets or queries the time-domain bandwidth filter.
[SENSe]:AVTime [:MEASview<y>]:{BANDwidth BWIDth}:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:AVTime [:MEASview<y>]:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:AVTime [:MEASview<y>]:MAXTracepoints	Sets or queries the maximum trace points.
[SENSe]:AVTime [:MEASview<y>]:METHod	Sets or queries the method to set the measurement bandwidth.
[SENSe]:AVTime [:MEASview<y>]:SPAN	Sets or queries the frequency span.
<b>[SENSe]:CCDF subgroup</b>	<b>CCDF measurement</b>
[SENSe]:CCDF:{BANDwidth BWIDth}	Sets or queries the CCDF measurement bandwidth.
[SENSe]:CCDF:CLEar	Clears the CCDF accumulator and restarts the measurement.
[SENSe]:CCDF:TIME:TOTal:LENGth	Sets or queries the CCDF measurement time.
[SENSe]:CCDF:TIME:TYPE	Sets or queries how to repeat the CCDF measurement.
[SENSe]:CONSte [:MEASview<y>]:SELEcted:SOURce	Sets or queries the Source of Constellation for a selected Measurement view number.
<b>[SENSe]:DDEMod subgroup (Option 21 only)</b>	<b>General purpose digital modulation measurements</b>
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth	Sets or queries the analysis length.
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:ACTual?	Queries the actual analysis length.
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:AUTO	Sets or queries whether to set the analysis length automatically.
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth BWIDth}:MODE	Sets or queries the measurement bandwidth mode.
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth BWIDth}:TINterval	Sets or queries the measurement bandwidth (frequency span).

Table continued...

Command	Description
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth BWIDth}:TINTerval:AUTO	Sets the measurement bandwidth (frequency span) automatically.
[SENSe]:DDEMod[:SOURce<x>]:BURSt:DETECT	Sets or queries how to detect bursts.
[SENSe]:DDEMod[:SOURce<x>]:BURSt:THReshold	Sets or queries the threshold level to determine a burst.
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet	Sets or queries the carrier frequency offset.
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet:AUTO	Sets or queries whether to detect the carrier frequency automatically.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:AVAIlable?	Determines if the equalizer is available.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:CONVergence	Sets or queries the Convergence value (also known as tap update rate).
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:ENABle	Sets or queries the Equalizer state.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:LENGth	Sets or queries the equalizer filter length.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:MODE	Sets or queries the equalizer mode. The equalizer can in learning mode (train) or enabled (hold).
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:RESet	Resets the equalizer settings to default values.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TAPs	Sets or queries the number of filter coefficients for the equalizer.
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TSRATio	Sets or queries the Equalizer Taps/Symbol parameter.
[SENSe]:DDEMod[:SOURce<x>]:FILTer:ALPHa	Sets or queries the filter factor (a/BT).
[SENSe]:DDEMod[:SOURce<x>]:FILTer:MEASurement	Sets or queries the measurement filter.
[SENSe]:DDEMod[:SOURce<x>]:FILTer:REFerence	Sets or queries the reference filter.
[SENSe]:DDEMod[:SOURce<x>]:FREQUency:DEViation	Sets or queries the frequency deviation to determine two states for FSK.
[SENSe]:DDEMod[:SOURce<x>]:FREQUency:DEViation:AUTO	Sets or queries whether to detect the frequency deviation automatically.
[SENSe]:DDEMod[:SOURce<x>]:MAGNitude:NORMalize	Sets or queries the method for the magnitude normalization.
[SENSe]:DDEMod[:SOURce<x>]:MINDEX	Sets or queries the modulation index for a CPM signal.
[SENSe]:DDEMod[:SOURce<x>]:MINDEX:AUTO	Sets or queries whether to detect the modulation index automatically.
[SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE	Sets or queries the modulation type.
[SENSe]:DDEMod[:SOURce<x>]:PRESet	Presets the modulation analysis to a communication standard.
[SENSe]:DDEMod[:SOURce<x>]:RING:RADIus	Sets or queries the APSK radius parameter.
[SENSe]:DDEMod[:SOURce<x>]:RING:ROTation	Sets or queries the APSK rotation parameter.
[SENSe]:DDEMod[:SOURce<x>]:SRATe	Sets or queries the symbol rate.
[SENSe]:DDEMod[:SOURce<x>]:SWAP:IQ	Sets or queries whether or not to swap I and Q data.
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:HSSHift	Sets or queries the Q data half-symbol shift for OQPSK and SOQPSK signals.
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP:SOURce?	Specifies or queries the user symbol map file.
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP:STATe]	Sets or queries whether to use the user symbol map.

Table continued...

Command	Description
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:PLOT:POSITION	Sets or queries the symbol point location on an SOQPSK waveform.
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:POINTS	Sets or queries the number of points per symbol.
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:RATE:SEARCH	Determines whether to enable a symbol rate search.
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD	Sets or queries whether to enable the synchronization word.
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD:SYMBOL	Sets or queries the synchronization word.
[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITS	Sets or queries the fundamental unit of time.
<b>[SENSe]:DIQVtime subgroup</b>	<b>Demodulated I&amp;Q versus time measurement view number</b>
[SENSe]:DIQVtime[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source for a selected Measurement view number.
<b>[SENSe]:DPX subgroup</b>	<b>DPX spectrum measurement</b>
[SENSe]:DPX:AUDio:DEMod:GAIN	Sets or queries the gain for the audio demodulation.
[SENSe]:DPX:AUDio:DEMod:RXBWidth	Sets or queries the receiver bandwidth for the audio demodulation.
[SENSe]:DPX:AUDio:DEMod:RXFRrequency?	Queries the receiver frequency for the audio demodulation.
[SENSe]:DPX:AUDio:DEMod:STATE	Sets or queries whether to enable or disable the audio demodulation.
[SENSe]:DPX:AUDio:DEMod:TUNE	Sets or queries how to determine the tuning frequency.
[SENSe]:DPX:AUDio:DEMod:TYPE	Sets or queries the modulation type for the audio demodulation.
[SENSe]:DPX:{BANDwidth BWIDth}:ACTual?	Sets or queries the resolution bandwidth (RBW).
[SENSe]:DPX:{BANDwidth BWIDth}:RESolution	Sets or queries the resolution bandwidth.
[SENSe]:DPX:{BANDwidth BWIDth}[:RESolution]:AUTO	Determines whether to set the resolution bandwidth (RBW) automatically or manually in the DPX spectrum measurement.
[SENSe]:DPX:FREQuency:SPAN: {BANDwidth BWIDth}[:RESolution]:RATio	Sets or queries whether to set the RBW automatically.
[SENSe]:DPX:CLEAr:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:DPX:COLor	Sets or queries the color palette of three-dimensional graphs.
[SENSe]:DPX:COLor:MAXimum	Sets or queries the maximum value of the color axis.
[SENSe]:DPX:COLor:MINimum	Sets or queries the minimum value of the color axis.
[SENSe]:DPX:DDENsity:MEASurement:SHOW	Sets or queries the Show measurement value (measurement always visible for DPX Trigger).
[SENSe]:DPX:DGRam:COLor:MAXimum	Sets or queries the maximum value of the color axis in the DPXogram measurement.
[SENSe]:DPX:DGRam:COLor:MINimum	Sets or queries the minimum value of the color axis in the DPXogram measurement.
[SENSe]:DPX:DGRam:POINTS:COUNT	Sets the number of trace points acquired for the DPXogram display.
[SENSe]:DPX:DGRam:TIME:CAPacity?	Returns the time capacity of the DPXogram measurement.
[SENSe]:DPX:DGRam:TIME:RESolution	Sets the time resolution of the DPXogram measurement.
[SENSe]:DPX:DGRam:TIME:RESolution:AUTO	Sets the time resolution automatically or manually in the DPXogram measurement.

Table continued...

Command	Description
[SENSe]:DPX:DGRam:WATERfall:DIRection	Sets or queries the direction of the Waterfall (DPXogram) display.
[SENSe]:DPX:DGRam:WATERfall:ENABle	Sets or queries the Waterfall (DPXogram) display.
[SENSe]:DPX:DGRam:Y[:SCALe]:AUTO	Rescales the height axis automatically to fit the waveform to the screen in the DPXogram display.
[SENSe]:DPX:DGRam:Y[:SCALe]:RESet	Resets the height scale of the DPXogram display.
[SENSe]:DPX:DWELI	Sets or queries the value of the Dwell time for the DPX spectrum measurement.
[SENSe]:DPX:DWELI:AUTO	Sets the value of the Dwell time automatically.
[SENSe]:DPX:PLOT	Sets or queries the plot in the DPX spectrum measurement.
[SENSe]:DPX:POINts:COUNT	Sets or queries the number of DPX trace points.
[SENSe]:DPX:FREQUency:CENTer	Sets or queries the center frequency.
[SENSe]:DPX:FREQUency:SPAN	Sets or queries the frequency span.
[SENSe]:DPX:FREQUency:STARt	Sets or queries the measurement start frequency.
[SENSe]:DPX:FREQUency:STEP	Sets or queries the frequency step size.
[SENSe]:DPX:FREQUency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:DPX:FREQUency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:DPX:TDM:{BANDwidth BWIDth}	Sets or queries the value of the measurement bandwidth in the DPX measurement.
[SENSe]:DPX:TDM:{BANDwidth BWIDth}:ACTual?	Queries the actual bandwidth in the DPX measurement.
[SENSe]:DPX:TDM:FREQUency:CENTer	Sets or queries the center frequency in the DPX spectrum measurement.
[SENSe]:DPX:TDM:FREQUency:TXBer:CLEar:RESults	Clears the measurement results of the active Tx BER measurement.
[SENSe]:DPX:TDM:FREQUency:TXBer:ENABle	Sets or queries the Tx BER measurement.
[SENSe]:DPX:TDM:FREQUency:TXBer:PATTern	Sets or queries the Tx BER data pattern.
[SENSe]:DPX:TDM:FREQUency:TXBer:PATTern:INVert	Sets or queries the data pattern Invert setting.
[SENSe]:DPX:TDM:FREQUency:TXBer:PATTern:REPeat	Sets or queries the test duration for a User Pattern.
[SENSe]:DPX:TDM:FREQUency:TXBer:PATTern:REVerse	Sets or queries whether the data pattern is in normal order or reversed.
[SENSe]:DPX:TDM:FREQUency:TXBer:PATTern: SOURce?	Returns the path and file name of the user-defined pattern file.
[SENSe]:DPX:TDM:RBW	Sets or queries the value of the RBW filter.
[SENSe]:DPX:TDM:RBW:STATe	Determines whether to set the resolution bandwidth (RBW) automatically or manually.
[SENSe]:DPX:TDM:SWEep:TIME	Sets or queries the sweep time.
[SENSe]:DPX:TDM:TMOtion	Sets or queries the scroll direction.
<b>[SENSe]:{FSETtling PSETtling} subgroup</b>	<b>Settling time measurements</b>
[SENSe]:{FSETtling PSETtling}:{BANDwidth BWIDTh}	Sets or queries the measurement bandwidth for the Settling Time measurements.

Table continued...

Command	Description
[SENSe]:{FSETtling PSETtling}: {BANDwidth BWIDTH}:ACTual?	Queries the actual measurement bandwidth in the Settling Time measurements.
[SENSe]:{FSETtling PSETtling}:FREQuency:CENTer	Queries the actual measurement bandwidth in the Settling Time measurements.
[SENSe]:{FSETtling PSETtling}:FREQuency:OFFSet	Sets or queries the offset frequency.
[SENSe]:{FSETtling PSETtling}:LENGth	Sets or queries the measurement length.
[SENSe]:{FSETtling PSETtling}:LENGth:ACTual?	Queries the actual measurement length for the Settling Time measurements.
[SENSe]:{FSETtling PSETtling}:MASK:LIMit	Sets or queries the frequency or phase limit for the specified region.
[SENSe]:{FSETtling PSETtling}:MASK:STATe	Sets or queries mask test state.
[SENSe]:{FSETtling PSETtling}:MASK:TIME:REFerence	Sets or queries the mask time reference.
[SENSe]:{FSETtling PSETtling}:MASK:TIME:START	Sets or queries the mask region start time in seconds.
[SENSe]:{FSETtling PSETtling}:MASK:TIME:STOP	Sets or queries the mask region stop time in seconds.
[SENSe]:{FSETtling PSETtling}:MAXTracepoints	Sets or queries the maximum trace points in the Settling Time measurements.
[SENSe]:{FSETtling PSETtling}:SDURation:MINimum	Sets or queries the minimum settled duration in seconds.
[SENSe]:{FSETtling PSETtling}:TARGet:REFerence	Sets or queries the target reference frequency.
[SENSe]:{FSETtling PSETtling}:TOLerance	Sets or queries the frequency or phase range into which the signal must settle.
<b>[SENSe]:FVTime subgroup</b>	<b>Frequency versus Time measurement</b>
[SENSe]:FVTime[:MEASview<y>]:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:FVTime[:MEASview<y>]:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:FVTime[:MEASview<y>]:FREQuency:START	Sets or queries the measurement start frequency.
[SENSe]:FVTime[:MEASview<y>]:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:FVTime[:MEASview<y>]:FREQuency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:FVTime[:MEASview<y>]:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:FVTime[:MEASview<y>]:MAXTracepoints	Sets or queries the maximum trace points.
[SENSe]:FVTime[:MEASview<y>]:SELEcted:SOURce	Sets or queries the Source with respect to the Frequency versus Time Measurement View number
<b>[SENSe]:IQVTime subgroup</b>	<b>RF I&amp;Q versus Time measurement</b>
[SENSe]:IQVTime[:MEASview<y>]:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:START	Sets or queries the measurement start frequency.
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:IQVTime[:MEASview<y>]:MAXTracepoints	Sets or queries the maximum trace points.

Table continued...

Command	Description
[SENSe]:IQVTime[:MEASview<y>]:SELEcted:SOURce	Sets or queries the Source with respect to the RF I&Q versus Time Measurement View number
<b>[SENSe]:MCPower subgroup</b>	<b>MCPR measurement</b>
[SENSe]:MCPower:AVERage	Sets or queries how to average waveform.
[SENSe]:MCPower:AVERage:COUNT	Sets or queries the number of waveforms for average.
[SENSe]:MCPower:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:MCPower:{BANDwidth BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:MCPower:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the RBW automatically.
[SENSe]:MCPower:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:MCPower:{BANDwidth BWIDth}:VIDeo:STATE	Sets or queries whether to enable the video bandwidth (VBW).
[SENSe]:MCPower:CHANnel:ADJacent:ADD	Adds a pair of upper and lower adjacent channels.
[SENSe]:MCPower:CHANnel:ADJacent:DELete	Deletes a selected adjacent channel.
[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>: ENABLE?	Queries whether the adjacent channel is selected or not in the P25 MCPR display.
[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>	Sets or queries the selected adjacent upper and lower limit values in the P25 MCPR display.
[SENSe]:MCPower:CHANnel:FILTer	Sets or queries the measurement filter.
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth BWIDth}	Sets or queries the frequency bandwidth of the main channels.
[SENSe]:MCPower:CHANnel:MAIN:COUNT	Sets or queries the number of main channels.
[SENSe]:MCPower:CHANnel:MAIN:INACTive	Makes a main channel inactive or queries the inactive main channels.
[SENSe]:MCPower:CHANnel:MAIN:SPACing	Sets or queries the main channel spacing.
[SENSe]:MCPower:CHIPrate	Sets or queries the chip rate.
[SENSe]:MCPower:CLEar:RESults	Restarts the average trace.
[SENSe]:MCPower:FREQuency	Sets or queries the center frequency.
[SENSe]:MCPower:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:MCPower:FREQuency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:MCPower:NFLoor:STATE	Sets or queries whether to enable the correction for noise floor.
[SENSe]:MCPower:POINts:COUNT	Sets or queries the number of trace points.
[SENSe]:MCPower:RCHannels?	Queries the power reference.
[SENSe]:MCPower:RCHannels:MAIN<x>	Sets the power reference to the main channel with the index (<x>).
[SENSe]:MCPower:RCHannels:TOTal	Sets the power reference to the total power of all the active channels.
[SENSe]:MCPower:RRCRolloff	Sets or queries the filter parameter for the Root Raised Cosine filter.
<b>[SENSe]:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
[SENSe]:OBWidth:AVERage	Sets or queries whether to enable or disable averaging.
[SENSe]:OBWidth:AVERage:COUNT	Sets or queries the number of measurements for averaging.
[SENSe]:OBWidth:{BANDwidth BWIDth}:MEASurement	Sets or queries the measurement bandwidth.

Table continued...



Command	Description
[SENSe]:OBWidth:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:OBWidth:{BANDwidth BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:OBWidth:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the RBW automatically.
[SENSe]:OBWidth:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:OBWidth:{BANDwidth BWIDth}:VIDeo:STATe	Sets or queries whether to enable the video bandwidth (VBW).
[SENSe]:OBWidth:CLear:RESults	Restarts the averaging.
[SENSe]:OBWidth:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:OBWidth:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:OBWidth:FREQuency:STEP:AUTO	Determines whether to set the frequency step size automatically.
[SENSe]:OBWidth:PERCent	Sets or queries the occupied bandwidth percent power.
[SENSe]:OBWidth:XDBLevel	Sets or queries the x dB level.
<b>[SENSe]:OFDM subgroup</b>	<b>Orthogonal Frequency Division Multiplexing measurement</b>
[SENSe]:OFDM:CHANnel:ESTimation	Sets or queries the channel estimation.
[SENSe]:OFDM:CHANnel[:BANDwidth BWIDth]	Sets or queries the channel bandwidth.
[SENSe]:OFDM:CHANnel[:BANDwidth BWIDth]:SRATe?	Returns the FFT sample rate.
[SENSe]:OFDM:CHANnel[:BANDwidth BWIDth]:STATe	Determines whether the channel bandwidth is on or off.
[SENSe]:OFDM:CONSte:DETermination	Sets or queries the constellation determination.
[SENSe]:OFDM:CONSte:DETermination:MANual	Sets or queries a manual constellation type.
[SENSe]:OFDM:FFT:LENGTh?	Returns the FFT length.
[SENSe]:OFDM:GUARd:INTerval	Sets or queries the guard interval.
[SENSe]:OFDM:GUARd:INTerval:USER	Sets or queries a user-defined guard interval.
[SENSe]:OFDM:PILot:TRACking:AMPLitude:STATe	Determines whether to select the Amplitude pilot tracking.
[SENSe]:OFDM:PILot:TRACking:PHASe:STATe	Determines whether to select the Phase pilot tracking.
[SENSe]:OFDM:PILot:TRACking:TIMing:STATe	Determines whether to select the Timing pilot tracking.
[SENSe]:OFDM:RADix	Sets the radix for the OFDM measurements.
[SENSe]:OFDM:SCARrier:SPACing	Sets or queries the subcarrier spacing.
[SENSe]:OFDM:SCARrier:SPACing:STATe	Determines whether the subcarrier spacing is on or off.
[SENSe]:OFDM:SCARriers	Sets or queries the Subcarriers to display.
[SENSe]:OFDM:SCARriers:RANGe:STARt	Sets or queries the subcarrier start range.
[SENSe]:OFDM:SCARriers:RANGe:STOP	Sets or queries the subcarrier stop range.
[SENSe]:OFDM:SCARriers:SINGle:INDex	Sets or queries the index of a single subcarrier.
[SENSe]:OFDM:STANdard	Sets or queries the OFDM standard.
[SENSe]:OFDM:SWAP:IQ	Determines whether or not to exchange I and Q data.
[SENSe]:OFDM:SYMBol:ANALysis:OFFSet	Sets or queries the OFDM symbol analysis offset.
[SENSe]:OFDM:SYMBols	Sets or queries the symbols to display.
[SENSe]:OFDM:SYMBols:MAX	Sets or queries the maximum number of symbols to analyze.
[SENSe]:OFDM:SYMBols:MAX:STATe	Enables or disables the maximum number of symbols to analyze.
Table continued...	

Command	Description
[SENSe]:OFDM:SYMBOLs:RANGe:COUNT	Sets or queries the number of symbols to display.
[SENSe]:OFDM:SYMBOLs:RANGe:START	Sets or queries the symbols start range.
[SENSe]:OFDM:SYMBOLs:SINGLe:INDEX	Sets or queries the symbol index
[SENSe]:OFDM:UNIT:FREQuency	Specifies the frequency units.
[SENSe]:OFDM:UNIT:TIME	Specifies the time units.
<b>[SENSe]:PHVTime subgroup</b>	<b>Phase versus Time measurement</b>
[SENSe]:PHVTime[:MEASview<y>]:CLEAr:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:START	Sets or queries the measurement start frequency.
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:PHVTime[:MEASview<y>]:MAXTracepoints	Sets or queries the maximum trace points.
[SENSe]:PHVTime[:MEASview<y>]:SELEcted:SOURce	Sets or queries the Source with respect to the Phaseversus Time Measurement View number
<b>[SENSe]:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurement</b>
[SENSe]:PNOise:AVERAge:COUNT	Sets or queries the number of traces to combine for averaging.
[SENSe]:PNOise:AVERAge:ENABLE	Sets or queries whether to enable or disable averaging trace.
[SENSe]:PNOise:CARRier:FREQuency:TRACk	Sets or queries whether to enable tracking the carrier frequency.
[SENSe]:PNOise:CARRier:THReshold	Sets or queries the threshold level to detect the carrier.
[SENSe]:PNOise:CLEAr:RESults	Restarts the average process.
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:START	Sets or queries the start offset frequency for integration.
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STOP	Sets or queries the stop offset frequency for integration.
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:START	Sets or queries the start offset frequency for plot.
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP	Sets or queries the stop offset frequency for plot.
[SENSe]:PNOise:OPTimization	Sets or queries the method of optimization.
<b>[SENSe]:PULSe subgroup</b>	<b>Pulsed RF measurements</b>
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:ALENght	Sets or queries the absolute amount of time included in the Frequency-Domain Linearity measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:METHod	Sets or queries the measurement time method for the Frequency-Domain Linearity measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLENght	Sets or queries the relative amount of time included in the Frequency-Domain Linearity measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLEVel	Sets or queries the Frequency-Domain Linearity measurement reference level.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:START	Sets or queries the start time for the Frequency-Domain Linearity measurements
Table continued...	

Command	Description
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:CORRectioN:AMPLitude[:STATe]	Enables or disables the impulse response measurement amplitude corrections for window and for delay of the highest side lobe.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:KOTime	Sets or queries the impulse response measurement Keep-out time.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:MEASurement:TIME:ALENgtH	Sets or queries the absolute amount of time included in impulse response measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:MEASurement:TIME:METhod	Sets or queries the impulse response measurement time method.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:MEASurement:TIME:RELENgtH	Sets or queries the relative amount of time included in Impulse Response measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:MEASurement:TIME:RLEVel	Sets or queries the measurement reference level for Impulse Response measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESPonse:MEASurement:TIME:STARt	Sets or queries the start time for the Impulse Response measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel	Sets or queries how to determine the 50% level.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:FIFTy	Sets or queries how to determine the 50% level.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:HUNDred	Sets or queries how to determine the 50% level.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:ALENgtH	Sets or queries the absolute amount of time included in the Overshoot measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:METhod	Sets or queries the measurement time method for the Overshoot measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RELENgtH	Sets or queries the relative amount of time included in the Overshoot measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLEVel	Sets or queries the measurement reference level for the Overshoot measurements.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:STARt	Sets or queries the start time for the Overshoot measurements
[SENSe]:PULSe[:MEASview<y>]:ANALyze:PMLocation	Sets or queries the phase measurement location.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:POINt:LOCation	Sets or queries the point location method.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RFALI	Sets or queries the threshold levels to measure the rise/fall time.
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RIPple	Sets or queries the ripple portion of the pulse top.
[SENSe]:PULSe[:MEASview<y>]:CARRier:FREQUency?	Queries the carrier frequency.
[SENSe]:PULSe[:MEASview<y>]:CARRier:OFFSet	Sets or queries the carrier frequency offset.
[SENSe]:PULSe[:MEASview<y>]:CARRier:SEARCh	Sets or queries how to detect the carrier.
[SENSe]:PULSe[:MEASview<y>]:DETEct:MEASurement	Sets or queries whether to set the maximum number of pulses.
[SENSe]:PULSe[:MEASview<y>]:DETEct:NUMBer	Sets or queries the maximum number of pulses to detect.
[SENSe]:PULSe[:MEASview<y>]:DETEct:POWer[:THReshold]	Sets or queries the power threshold to detect pulses.
[SENSe]:PULSe[:MEASview<y>]:DETEct:TIMe[:THReshold]	Sets or queries the minimum off-time between pulses.
[SENSe]:PULSe[:MEASview<y>]:FILTer:{BANDwidth BWIth}	Sets or queries the filter bandwidth for the Gaussian filter.
[SENSe]:PULSe[:MEASview<y>]:FILTer:MEASurement	Sets or queries the measurement filter.

Table continued...

Command	Description
[SENSe]:PULSe[:MEASview<y>]:FREference:AUTO	Determines whether to estimate the frequency reference automatically.
[SENSe]:PULSe[:MEASview<y>]:FREference:CSlope	Sets or queries the Chirp Slope.
[SENSe]:PULSe[:MEASview<y>]:FREference:CSlope:AUTO	Sets or queries enabling automatic Chirp Slope estimating.
[SENSe]:PULSe[:MEASview<y>]:FREference:OFFSet	Sets or queries the frequency reference offset.
[SENSe]:PULSe[:MEASview<y>]:MODulation:TYPE	Sets or queries the modulation type.
[SENSe]:PULSe[:MEASview<y>]:SIGNal:TYPE	Sets or queries the signal type in the pulsed RF measurements.
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:ORDinate	Sets or queries the Histogram Y unit.
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINts	Sets or queries the number of bins in the Histogram plot.
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINts:AUTO	Sets or queries enabling automatic Histogram plot bins count.
<b>[SENSe]:REANalyze subgroup</b>	<b>Spectrogram measurement</b>
[SENSe]:REANalyze	Sets all measurements to reanalyze the current acquisition record.
[SENSe]:REANalyze:ACQuisition:SETTings?	Queries the settings of all acquisitions in acquisition history.
[SENSe]:REANalyze:ALL	Replay all items in the current acquisition history selection.
[SENSe]:REANalyze:ALL:LOOP	Replay all selected records/frames in a continuous loop.
[SENSe]:REANalyze:CURRent	Replay the currently selected acquisition (or frame).
[SENSe]:REANalyze:CURRent:ACQuisition?	Queries the Acquisition record identifier (ID) of the currently selected acquisition (or frame).
[SENSe]:REANalyze:CURRent:FRAME?	Queries the Frame identifier (ID) of the currently selected acquisition (or frame).
[SENSe]:REANalyze:CURRent:TIMestamp?	Queries the time stamp of the currently selected acquisition (or frame).
[SENSe]:REANalyze:DELeTe:DATA	Deletes all acquisition data.
[SENSe]:REANalyze:FIRSt	Replays the first item in the current acquisition history selection.
[SENSe]:REANalyze:LAST	Replays the last item in the current acquisition history selection.
[SENSe]:REANalyze:NEXt	Replays the next item in the current acquisition history selection.
[SENSe]:REANalyze:PAUSe	Sets or queries the reanalysis pause state.
[SENSe]:REANalyze:PREVious	Replays the previous item in the current Replay selection.
[SENSe]:REANalyze:SElect:ACQuisition:DATA:TYPE	Sets or queries the acquisition data type.
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt?	Queries the first selectable acquisition history Start or Stop value.
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt: TIMestamp?	Queries the time stamp associated with the first selectable acquisition history Start or Stop value.
[SENSe]:REANalyze:SElect:ACQuisition:LAST?	Queries the last selectable acquisition history Start or Stop value
[SENSe]:REANalyze:SElect:ACQuisition:LAST: TIMestamp?	Queries the time stamp associated with the last selectable acquisition history Start or Stop value
[SENSe]:REANalyze:SElect:ACQuisition:START	Sets or queries the Acquisition Start value in the acquisition history selection.

Table continued...

Command	Description
[SENSe]:REANalyze:SElect:ACQuisition:START: TIMestamp?	Queries the time stamp associated with the current Acquisition Start selection
[SENSe]:REANalyze:SElect:ACQuisition:STOP	Sets or queries the Acquisition Stop value in the acquisition history selection.
[SENSe]:REANalyze:SElect:ACQuisition:STOP: TIMestamp?	Queries the time stamp associated with the current Acquisition Stop selection.
[SENSe]:REANalyze:SElect:ALL	Selects all acquisitions and frames in the current acquisition history.
[SENSe]:REANalyze:SElect:FRAME:FIRSt?	Queries the first available Frame Start or Stop value available for acquisition history selection.
[SENSe]:REANalyze:SElect:FRAME:FIRSt: TIMestamp?	Queries the time stamp associated with the first available frame of the first available Acquisition Start or Stop value
[SENSe]:REANalyze:SElect:FRAME:LAST?	Queries the last available Frame Start or Stop value available for acquisition history selection.
[SENSe]:REANalyze:SElect:FRAME:LAST: TIMestamp?	Queries the time stamp associated with the last available frame of the last available Acquisition Start or Stop value
[SENSe]:REANalyze:SElect:FRAME:START	Sets or queries the Frame Start value in the acquisition history selection.
[SENSe]:REANalyze:SElect:FRAME:START: TIMestamp?	Queries the time stamp associated with the current Frame Start selection.
[SENSe]:REANalyze:SElect:FRAME:STOP	Sets or queries the frame stop value in acquisition history selection.
[SENSe]:REANalyze:SElect:FRAME:STOP: TIMestamp?	Queries the time stamp associated with the current frame stop selection.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:ALL	Selects all spectra acquisitions and frames in the currently selected set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: CURRent:ACQuisition?	Queries the acquisition record identifier (ID) of the currently selected acquisition.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: CURRent:TIMestamp?	Queries the time stamp of the currently selected acquisition.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: FIRSt?	Queries the first selectable acquisition start or stop value in the currently selected set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: FIRSt:TIMestamp?	Queries the time stamp of the first selectable acquisition history start or stop value in the currently selected acquisition set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: LAST?	Queries the last selectable acquisition start or stop value in the currently selected set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: LAST:TIMestamp?	Queries the time stamp of the last selectable acquisition start or stop value in the currently selected set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: START	Sets or queries the acquisition start value in the currently selected set.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition: START:TIMestamp?	Queries the time stamp of the current acquisition start selection.
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP	Sets or queries the acquisition stop value in the currently selected set.

Table continued...

Command	Description
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP:TIMestamp?	Queries the time stamp of the current acquisition stop selection.
[SENSe]:REANalyze:SPEed	Sets or queries the time delay value inserted between reanalysis of frames or acquisitions.
[SENSe]:REANalyze:STOP	Stops the current reanalysis.
[SENSe]:REANalyze:TIMestamp:DECimal	Sets or queries the number of values to the right of the decimal point to include in all time stamps.
<b>[SENSe]:SEM subgroup</b>	<b>Spectrum Emissions Mask measurements</b>
[SENSe]:SEM:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).
[SENSe]:SEM:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:SEM:{BANDwidth BWIDth}:VIDeo:STATe	Sets or queries the state of the video bandwidth (VBW) of the reference channel.
[SENSe]:SEM:CHANnel:FILTer	Sets or queries the reference Channel Filter.
[SENSe]:SEM:CHANnel:IBANDwidth	Sets or queries the reference channel integration bandwidth.
[SENSe]:SEM:CHANnel:WIDTh	Sets or queries the reference Channel Filter bandwidth.
[SENSe]:SEM:CHIPrate	Sets or queries the Chip rate.
[SENSe]:SEM:CLEar:RESults	Clears the results in the Spectral Emissions Mask measurement.
[SENSe]:SEM:FREQuency	Sets or queries the measurement frequency.
[SENSe]:SEM:FREQuency:SPAN?	Returns the frequency span.
[SENSe]:SEM:MFILTer	Sets or queries the shape of the filter.
[SENSe]:SEM:MODE:REALtime	Sets or queries the Real-time SEM measurement mode.
[SENSe]:SEM:MTYPE	Sets or queries the measurement type.
[SENSe]:SEM:OFFSet:DETection	Sets or queries the detection settings for all offsets.
[SENSe]:SEM:OFFSet<x>:BANDwidth:FILTer?	Returns the bandwidth filter setting.
[SENSe]:SEM:OFFSet<x>:BANDwidth[:RESolution]	Sets or queries the resolution bandwidth.
[SENSe]:SEM:OFFSet<x>:BANDwidth:RESolution:COUNT	Sets or queries the count of Resolution Bandwidths (RBWs).
[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo	Sets or queries the video bandwidth setting.
[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo:STATe	Sets or queries the state of the video bandwidth.
[SENSe]:SEM:OFFSet<x>:FREQuency:START	Sets or queries the start frequency.
[SENSe]:SEM:OFFSet<x>:FREQuency:STOP	Sets or queries the stop frequency.
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:SAME	Sets or queries the state of the limits for the absolute start and stop settings.
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:START	Sets or queries the absolute power start limit.
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:STOP	Sets or queries the absolute power stop limit.
[SENSe]:SEM:OFFSet<x>:LIMIT:MASK	Sets or queries the limit mask setting.
[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:SAME	Sets or queries the state of the limits for the relative start and stop settings
[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:START	Sets or queries the relative power start limit.
[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:STOP	Sets or queries the relative power stop limit.

Table continued...

Command	Description
[SENSe]:SEM:OFFSet<x>:LIMIT:SIDE	Sets or queries the side of the offset limit.
[SENSe]:SEM:OFFSet<x>:STATe	Determines whether to display the specified row of the Offsets & Limits table.
[SENSe]:SEM:POINTs:COUNT	Sets or queries the maximum number of trace points.
[SENSe]:SEM:PRESet	Defines the standard protocol settings for the measurement.
[SENSe]:SEM:REFerence:AUTO	Determines whether to automatically update the Power Reference.
[SENSe]:SEM:REFerence:DETection	Sets or queries the reference channel detection setting.
[SENSe]:SEM:REFerence[:MAGNitude]	Sets or queries the power magnitude.
[SENSe]:SEM:RRCRolloff	Sets or queries the filter parameter (roll-off ratio).
[SENSe]:SEM:STEP:AUTO	Determines whether to automatically update the center frequency.
[SENSe]:SEM:STEP[:MAGNitude]	Sets or queries the step magnitude.
[SENSe]:SEM:WIDTH:DEFinition	Sets or queries the offset definition.
<b>[SENSe]:SGRam subgroup</b>	<b>Spectrogram measurements</b>
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}:RESolution	Sets or queries the resolution bandwidth (RBW).
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the RBW automatically.
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:MODE	Sets or queries whether to enable or disable the RBW processing.
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth BWIDth}:VIDeo:STATe	Sets or queries whether to enable the video bandwidth (VBW).
[SENSe]:SGRam[:MEASview<y>]:COLor	Sets or queries the color palette of three-dimensional graphs.
[SENSe]:SGRam[:MEASview<y>]:COLor:MAXimum	Sets or queries the maximum value of the color axis.
[SENSe]:SGRam[:MEASview<y>]:COLor:MINimum	Sets or queries the minimum value of the color axis.
[SENSe]:SGRam[:MEASview<y>]:FFT:WINDow	Sets or queries the FFT window.
[SENSe]:SGRam[:MEASview<y>]:FILTer[:SHAPE]	Sets or queries the filter shape.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:CENTer	Sets or queries the center frequency.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:SPAN	Sets or queries the frequency span.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:SPAN:BANDwidth[:RESolution]:RATio	Sets or queries the ratio of span to RBW.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:SPAN:MAXimum	Sets the frequency range to the maximum real-time span.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:STARt	Sets or queries the measurement start frequency.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:STEP	Sets or queries the frequency step size.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:SGRam[:MEASview<y>]:FREQUency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:SGRam[:MEASview<y>]TIME[:SCALE]:MODE	Sets or queries the vertical mode.

Table continued...

Command	Description
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:OVERlap:PERCent	Sets or queries the Overlap in percent.
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:PER:DIVision	Sets or queries the vertical Time/division value.
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:PER:UPDate:MINutes	Sets or queries the Spectrum Monitor mode Time/Update value for minutes.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:ENABle	Sets or queries the Waterfall (3-D spectrogram) display.
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:PER:UPDate:SEConds	Sets or queries the Spectrum Monitor mode Time/Update value seconds
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:SPECtrums:PERLine?	Queries the Spectrums/Line value.
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:STARt:DIVisions	Sets or queries the Vertical offset or Position in divisions.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:DIRection	Sets or queries the direction of the Waterfall display.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y:AUTO	Rescales the height axis automatically to fit the waveform to the screen in the 3-D Waterfall display.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y:OFFSet	Sets or queries the height offset (the value at the top edge of the vertical axis) of the 3-D Waterfall display.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y:RESet	Resets the height scale of the 3-D Waterfall display to the default values.
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALe]	Sets or queries the height (vertical) range of the Waterfall (3-D spectrogram) display.
<b>[SENSE]:SIGNAlvu subgroup</b>	<b>Signalvu measurement</b>
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:COUPling	Sets or queries the per channel coupling of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:DESKew	Sets or queries the per channel deskew of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:EXTattn	Sets or queries the per channel external attenuation of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel?:INSTrument:BWIDth	Queries the per channel instrument bandwidth of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel?:REFlevel	Sets or queries per channel reference level of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:PER:CHANnel?:TERMination	Sets or queries the per channel Termination of the measurement.
[SENSe]:SIGNAlvu:ACQuisition:CONTRol:SOURce<x>:STATe	Sets or queries the Source checkbox state of measurement source for RF, I and Q, Diff I and Q data type.
[SENSe]:SIGNAlvu:ACQuisition:GLOBal:REFlevel	Sets or queries the Global Reference level of the measurement source.
[SENSe]:SIGNAlvu:ACQuisition:INPuttype:REFlevel	Sets or queries the input type of ref level of the measurement source.
[SENSe]:SIGNAlvu:ACQuisition:NULL:IQ:OFFSet:CLEAR	To clear Null IQ offset values that were calculated using Calibrate IQ, use the above PI command.
[SENSe]:SIGNAlvu:ACQuisition:SELEcted:CHANnel:TYPE	Sets or queries the selected channel type of measurement.
[SENSe]:SIGNAlvu:ACQuisition:CONTRol:HORIZontal	Sets or queries the Acquisition or Horizontal settings. When the checkbox is checked, the SignalVu controls horizontal settings of the oscilloscope.

Table continued...



Command	Description
[SENSe]:SIGNALvu:ACQuisition:CONTRol:SAMPle:RATE	Sets or queries the Sample rate when the checkbox is checked, the SignalVu controls Sample rate of the oscilloscope.
[SENSe]:SIGNALvu:ACQuisition:CONTRol:TRIGger:POSition	Sets or queries the Trigger position, when checkbox is checked, the SignalVu controls Trigger position of the oscilloscope.
[SENSe]:SIGNALvu:ACQuisition:CONTRol:VERTical	Sets or queries the Vertical settings, when the checkbox is checked, the SignalVu controls the vertical settings of the oscilloscope.
[SENSe]:SIGNALvu:ACQuisition:DIgitizer:SAMPle:RATE	Sets or queries the oscilloscope sample rate from the Acquisition control panel oscilloscope settings of SignalVu.
[SENSe]:SIGNALvu:ACQuisition:DIgitizer:SAMPle:RATE:AUTO	Sets the oscilloscope sample rate automatically or manually from the Acquisition control panel oscilloscope settings of SignalVu.
[SENSe]:SIGNALvu:ACQuisition:IQ:CALibrate	Starts the IQ calibration.
[SENSe]:SIGNALvu:ACQuisition:RESet	Clears the acquisition data and restarts the SignalVu acquisitions from the oscilloscope.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CALibrate:IQ	For performing null IQ offset operation for each source use the above-mentioned PI command. It will be effective from the next acquisition.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:I	Sets or queries the channel of measurement source for I data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:IMINus	Sets or queries the channel of measurement source for IMinus data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:IPLus	Sets or queries the channel of measurement source for IPlus data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:Q	Sets or queries the channel of measurement source for Q data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:QMINus	Sets or queries the channel of measurement source for QMinus data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel:QPLus	Sets or queries the channel of measurement source for QPlus data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:CHANnel[:RF]	Sets or queries the channel of measurement source for RF data type.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJECTIONSide	Sets or queries the downconverter injection side.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:DOWNconverter:INSTRument:BANDwidth:MAXFreq	Sets or queries the maximum range of instrument bandwidth.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:DOWNconverter:INSTRument:BANDwidth:MINFreq	Sets or queries the minimum range of instrument bandwidth.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:DOWNconverter:LOFFreq	Sets or queries the downconverter low frequency.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:DOWNconverter:STATE	Sets or queries the downconverter state.
[SENSe]:SIGNALvu:ACQuisition[:SOURce<x>]:PERSrc:REFlevel	Sets or queries the Reference level Per source.
<b>[SENSe]:SPECTrum subgroup</b>	<b>Spectrum measurement</b>
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWidth}[:RESolution]	Sets or queries the resolution bandwidth (RBW).

Table continued...

Command	Description
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:ACTual?	Queries the actual resolution bandwidth (RBW).
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the RBW automatically.
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWIDth}[:RESolution]:MODE	Sets or queries whether to enable or disable the RBW process.
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWIDth}:VIDeo	Sets or queries the video bandwidth (VBW).
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth BWIDth}:VIDeo:STATe	Sets or queries whether to enable the video bandwidth (VBW).
[SENSe]:SPECTrum[:MEASview<y>]:CLEar:RESults	Restarts multi-trace functions (Average and Max/Min Hold).
[SENSe]:SPECTrum[:MEASview<y>]:FFT:WINDow	Sets or queries the FFT window.
[SENSe]:SPECTrum[:MEASview<y>]:FILTer[:SHAPE]	Sets or queries the filter shape.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN	Sets or queries the frequency span.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATIo	Sets or queries the ratio of span to RBW.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START	Sets or queries the measurement start frequency.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP	Sets or queries the frequency step size.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO	Sets or queries whether to set the frequency step size automatically.
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP	Sets or queries the measurement stop frequency.
[SENSe]:SPECTrum[:MEASview<y>]:LENGth	Sets or queries the spectrum length.
[SENSe]:SPECTrum[:MEASview<y>]:LENGth:ACTual?	Queries the actual spectrum length.
[SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO	Sets or queries the spectrum length mode (automatic or manual).
[SENSe]:SPECTrum[:MEASview<y>]:MAX:SPAN	Sets the frequency span to the maximum span.
[SENSe]:SPECTrum[:MEASview<y>]:POINts:COUNT	Sets or queries the number of sample points on the signal spectrum.
[SENSe]:SPECTrum[:MEASview<y>]:START	Sets or queries the spectrum offset time.
[SENSe]:SPECTrum[:MEASview<y>]:START:AUTO	Sets or queries the spectrum offset mode (automatic or manual).
[SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE	Sets or queries the spectrum time parameter mode (automatic or manual).
<b>[SENSe]:SPURious subgroup</b>	<b>Spurious measurement</b>
[SENSe]:SPURious:CARRier:{BANDwidth BWIDth}	Sets or queries the channel width for the carrier as power reference.
[SENSe]:SPURious:CARRier:{BANDwidth BWIDth}:INTegration	Sets or queries the integration bandwidth to calculate the carrier power.
[SENSe]:SPURious:CARRier:{BANDwidth BWIDth}[:RESolution]	Sets or queries the resolution bandwidth to measure the carrier power.
[SENSe]:SPURious:CARRier:{BANDwidth BWIDth}[:RESolution]:AUTO	Sets or queries whether to set the resolution bandwidth automatically.
[SENSe]:SPURious:CARRier:DETection	Sets or queries the carrier detection method.
[SENSe]:SPURious:CARRier:FREQuency	Sets or queries the carrier frequency.

Table continued...

Command	Description
[SENSe]:SPURious:CARRier:THReshold	Sets or queries the threshold level to detect the carrier.
[SENSe]:SPURious:CLear:RESults	Restarts multi-trace functions (Average and Max Hold).
[SENSe]:SPURious[:FREQuency]:OVERlap?	Queries whether any of the frequency ranges (A to T) overlap.
[SENSe]:SPURious:LIST	Sets or queries how to list the spurious signals.
[SENSe]:SPURious:MODE	Sets or queries the frequency range mode (Multi or Single).
[SENSe]:SPURious:OPTimization	Sets or queries the method of optimization.
[SENSe]:SPURious:POINts:COUNT	Sets or queries the trace point count for the frequency range.
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo	Sets or queries the VBW for the specified frequency range.
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe	Sets or queries whether to enable the VBW for the frequency range.
[SENSe]:SPURious:RANGe<x>:DETEction	Sets or queries the display detector for the frequency range.
[SENSe]:SPURious:RANGe<x>:EXCURsion	Sets or queries the excursion level in the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]	Sets or queries the filter shape for the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth	Sets or queries the filter bandwidth for the frequency range.
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPe]:BANDwidth:AUTO	Sets or queries whether to set the filter bandwidth automatically.
[SENSe]:SPURious:RANGe<x>:FREQuency:START	Sets or queries the start frequency of the range.
[SENSe]:SPURious:RANGe<x>:FREQuency:STOP	Sets or queries the stop frequency of the range.
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START	Sets or queries the absolute start amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP	Sets or queries the absolute stop amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:MASK	Sets or queries the limit mask function mode for the frequency range.
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:START	Sets or queries the relative start amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP	Sets or queries the relative stop amplitude of the limits for the range.
[SENSe]:SPURious:RANGe<x>:STATe	Sets or queries whether to enable or disable the frequency range.
[SENSe]:SPURious:RANGe<x>:THReshold	Sets or queries the threshold level to detect spurious signals in a range.
[SENSe]:SPURious:REFerence	Sets or queries the power reference.
[SENSe]:SPURious:REFerence:MANual:POWer	Sets or queries the reference power level.
<b>[SENSe]:SQUality subgroup</b>	<b>Signal quality</b>
[SENSe]:SQUality[:MEASview<y>]:EVM:ALL[:STATe]	Sets or queries the display of Offset EVM results.
<b>[SENSe]:SVPC subgroup</b>	<b>Channel trigger</b>
[SENSe]:SVPC:ACQuisition:CONTRol:CHANnel	Sets or queries the Channel Settings checkbox state.
[SENSe]:SVPC:ACQuisition:CONTRol:TRIGger	Sets or queries the Trigger Settings checkbox state.
<b>[SENSe]:TOVerview subgroup</b>	<b>Time overview</b>
[SENSe]:TOVerview[:MEASview<y>]:FREQuency:CENTer	Sets or queries the center frequency.
[SENSe]:TOVerview[:MEASview<y>]:MAXTracepoints	Sets or queries the maximum trace points.

Table continued...

Command	Description
[SENSe]:TOVerview[:MEASview<y>]:TIME:PER:DIVision	Sets or queries the time per division (dB/div) value in the Time Overview display.
<b>[SENSe]:USETtings subgroup</b>	<b>All measurements</b>
[SENSe]:USETtings	Updates the analyzer settings.
<b>[SENSe]:WLAN subgroup</b>	<b>Wireless LAN settings</b>
[SENSe]:WLAN:ANALysis:LENGth	Specifies the length of the analysis period.
[SENSe]:WLAN:ANALysis:LENGth:ACTual?	Returns the value of the actual analysis length.
[SENSe]:WLAN:ANALysis:LENGth:AUTO	Sets the analysis length value based on the requirements of the selected display.
[SENSe]:WLAN:ANALysis:OFFSet	Specifies or queries the analysis offset in the symbol interval.
[SENSe]:WLAN:BURSt	Specifies the burst to measure when multiple bursts are present in an acquisition.
[SENSe]:WLAN:CHANnel[:BANDwidth]:BWiDth]	Specifies or queries a nominal channel bandwidth to use for the WLAN measurements.
[SENSe]:WLAN:EQUalizer:TRAIning	Specifies the method for estimating channel frequency response and equalization.
[SENSe]:WLAN:FFT:LENGth?	Returns the fixed FFT length for the WLAN measurements.
[SENSe]:WLAN:FFT:SRATe?	Returns the FFT sample rate for the WLAN measurements.
[SENSe]:WLAN:FREQuency:ERRor	Sets or queries the frequency error in the WLAN analysis parameters.
[SENSe]:WLAN:FREQuency:ERRor:AUTO	Turns on the automatic determination of the frequency error.
[SENSe]:WLAN:GUARd:INTerval	Sets or queries the Guard Interval to use in the input signal.
[SENSe]:WLAN:MODulation:DETEct:QAM1024	Specifies a fixed modulation type.
[SENSe]:WLAN:MODulation:SOURce	Specifies the method of data symbol modulation identification.
[SENSe]:WLAN:PILot:TRACking:AMPLitude:STATe	Specifies the pilot subcarrier to use to correct amplitude variations over the packet.
[SENSe]:WLAN:PILot:TRACking:PHASe:STATe	Specifies the pilot subcarrier to use to correct phase variations over the packet.
[SENSe]:WLAN:PILot:TRACking:TIMing:STATe	Specifies the pilot subcarrier to use to correct timing variations over the packet.
[SENSe]:WLAN:RADix	Specifies or queries the format for displaying the symbol values.
[SENSe]:WLAN:SCARriers	Sets or queries which subcarriers to display in the WLAN view.
[SENSe]:WLAN:SCARriers:RANGe:STARt	Sets or queries the start value of the subcarrier range.
[SENSe]:WLAN:SCARriers:RANGe:STOP	Sets or queries the stop value of the subcarrier range.
[SENSe]:WLAN:SCARriers:SINGle:INDex	Specifies or queries the single subcarrier to display in the WLAN view.
[SENSe]:WLAN:SCARriers:SPACing	Sets or queries the subcarrier spacing.
[SENSe]:WLAN:SCARriers:SPACing:AUTO	Automatically sets the subcarrier spacing.
[SENSe]:WLAN:SIGNal:STANdard:TYPE	Sets or queries the WLAN signal standard.
Table continued...	

Command	Description
[SENSe]:WLAN:SubCARrier:DERotation	Shows subcarriers for displays with or without the Gamma subcarrier phase rotation removed.
[SENSe]:WLAN:SUMMary:CLEar:RESults	Clears the results of the WLAN Summary display.
[SENSe]:WLAN:SUMMary:CLEar:RESults	Clears the results in the WLAN Summary measurement.
[SENSe]:WLAN:SUMMary:EVM:RMS:AVERage: COUNT	Specifies the maximum number of bursts to average.
[SENSe]:WLAN:SUMMary:EVM:RMS:AVERage: COUNT:ENABLE	Enables the burst average function.
[SENSe]:WLAN:SWAP:IQ	Specifies to swap the I and Q components of a signal.
[SENSe]:WLAN:SYMBols CHIPs	Specifies or queries the symbols or chips are displayed in the WLAN Symbol Table.
[SENSe]:WLAN:SYMBols CHIPs:MAX	Sets or queries the value for the maximum number of symbols/chips to analyze.
[SENSe]:WLAN:SYMBols CHIPs:MAX:STATe	Turns the maximum data symbols to analyze feature on or off.
[SENSe]:WLAN:SYMBols CHIPs:RANGe:COUNT	Sets or queries the number of symbols or chips to display.
[SENSe]:WLAN:SYMBols CHIPs:RANGe:START	Sets or queries the symbols/chips start value.
[SENSe]:WLAN:SYMBols CHIPs:RANGe:STOP	Sets or queries the symbols or chips stop value.
[SENSe]:WLAN:SYMBols CHIPs:SINGle:INDEX	Sets or queries the symbol or chip to use when displaying results from a single symbol.
[SENSe]:WLAN:UNIT:FREQuency	Specifies or queries whether the displayed frequency units are frequency (Hz) or subcarrier.
[SENSe]:WLAN:UNIT:TIME	Specifies or queries whether the displayed time units are seconds, symbols or chips.
<b>SENSe:LTE subgroup</b>	
SENSe:LTE:ACLR:BANDwidth:RESolution	Sets or queries the resolution bandwidth (RBW).
SENSe:LTE:ACLR:BANDwidth:VIDeo	Sets or queries the video bandwidth (VBW).
SENSe:LTE:ACLR:BANDwidth:VIDeo:STATe	Sets or queries enabling or disabling the video bandwidth (VBW).
SENSe:LTE:ACLR:CHANnel:IBANDwidth	Sets or queries the reference channel integration bandwidth.
SENSe:LTE:ACLR:CHANnel:WIDTh	Sets or queries the reference channel width.
SENSe:LTE:ACLR:CLEar:RESults	Clears the results in the measurement.
SENSe:LTE:ACLR:FREQuency	Sets or queries the measurement center frequency.
SENSe:LTE:ACLR:MFILter	Sets or queries the shape of the filter.
SENSe:LTE:ACLR:MODE:REALtime	Sets or queries the real-time LTE ACLR measurement mode.
SENSe:LTE:ACLR:NFLoor:ACQuire	Enables measurement of the noise floor.
SENSe:LTE:ACLR:NFLoor:CORReCted?	Returns whether or not noise correction was applied on the last acquisition.
SENSe:LTE:ACLR:NFLoor:STATe	Determines whether to enable or disable the correction for noise floor.
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution	Sets or queries the resolution bandwidth (RBW) value in the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:COUNT	Sets or queries the count for the resolution bandwidths in the specified row in the Offsets & Limits table.

Table continued...

Command	Description
SENSe:LTE:ACLR:OFFSet<X>:BANDwidth:RESolution:FILTer	Sets or queries the filter bandwidth resolution under Offset & Limit table.
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo	Sets or queries the resolution video bandwidth (VBW) value in the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo:STATe	Sets or queries whether to enable or disable the video resolution bandwidth (VBW) in the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:DETectioN	Sets or queries the detection settings for all offsets.
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:BANDwidth	Sets or queries the frequency bandwidth value for the specified row of the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:OFFSet	Sets or queries the frequency offset value of the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:LIMit:ABSolute	Sets or queries the absolute limit value of the row in the Offset & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:LIMit:MASK	Sets or queries the limit mask setting for the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:LIMit:RELative	Sets or queries the relative limit value of channel under Offset & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:LIMit:SIDE	Sets or queries the side of the offset limit for the specified row in the Offsets & Limits table.
SENSe:LTE:ACLR:OFFSet<x>:STATe	Sets or queries the specified row of the Offsets & Limits table.
SENSe:LTE:ACLR:POINts:COUNT	Sets or queries the number of points used for the trace.
SENSe:LTE:ACLR:REFerence:MAGNitude:AUTO	Sets or queries the reference magnitude to automatic or manual.
SENSe:LTE:ACLR:REFerence:MAGNitude:DETectioN	Sets or queries the reference magnitude detection.
SENSe:LTE:ACLR:REFerence:MAGNitude	Sets or queries the reference power magnitude.
SENSe:LTE:ACLR:STEP:AUTO	Determines whether to automatically update the center frequency increment/decrement step size.
SENSe:LTE:ACLR:STEP:MAGNitude	Sets or queries the step magnitude for the center frequency.
SENSe]:LTE:ANALysis:EQUAlization:STATe	Sets or queries the Enable Equalization state in the LTE Analysis Params tab.
SENSe:LTE:ANALysis:LENGth:ACTual?	Queries the actual analysis length for the LTE Constellation and LTE Power vs Time displays.
[SENSe]:LTE:ANALysis:OFFSet	Sets or queries the LTE analysis offset for the LTE Constellation or LTE Power vs Time.
SENSe:LTE:ANALysis:OFFSet:AUTO	Sets or queries the Auto checkbox state for analysis offset.
SENSe:LTE:ANALysis:TIME:UNIT	Sets or queries the unit preference under the LTE Analysis tab.
SENSe:LTE:CHANnel:BANDwidth	Specifies or queries the channel bandwidth to use.
SENSe:LTE:CHSPectrum:AVERAge	Sets or queries how to average the waveform.
SENSe:LTE:CHSPectrum:AVERAge:COUNT	Sets or queries the number of traces for averaging.
SENSe:LTE:CHSPectrum:BANDwidth:RESolution	Sets or queries the resolution bandwidth (RBW).
SENSe:LTE:CHSPectrum:BANDwidth:RESolution:AUTO	Determines whether to set the resolution bandwidth (RBW) automatically or manually. Queries this setting.

Table continued...

Command	Description
SENSe:LTE:CHSPectrum:BANDwidth:VIDeo	Sets or queries the video bandwidth (VBW).
SENSe:LTE:CHSPectrum:BANDwidth:VIDeo:STATe	Sets or queries enabling or disabling the video bandwidth (VBW).
SENSe:LTE:CHSPectrum:CHANnel:BANDwidth	Specifies or queries a nominal channel bandwidth to use.
SENSe:LTE:CHSPectrum:CLEar:RESults	Clears the results in the LTE Channel Spectrum measurement.
SENSe:LTE:CHSPectrum:DETection	Sets or queries the trace detection settings.
SENSe:LTE:CHSPectrum:FREQuency	Sets or queries the measurement center frequency.
SENSe:LTE:CHSPectrum:FREQuency:SPAN	Sets or queries the frequency span.
SENSe:LTE:CHSPectrum:FREQuency:STEP	Sets or queries the frequency step size.
SENSe:LTE:CHSPectrum:FREQuency:STEP:AUTO	Determines whether to set the frequency step size automatically or manually. Queries this setting.
SENSe:LTE:CHSPectrum:POINts:COUNT	Sets or queries the number of points used for the trace.
SENSe:LTE:FRAME:STRUcture	Sets or queries the frame structure.
<b>[SENSe]: TDDiagram</b>	<b>TDDiagram measurement</b>
[SENSe]:TDDiagram[:MEASview<y>]:SELEcted:SOURce	Sets or queries the source of Trellis diagram for a selected Measurement view number.
SENSe:TOFF:POWEr:SPECTral:DENSity	Sets or queries the limit set for comparison of the Toff power spectral density value.

## Status commands

Use the STATUS commands to control registers defined in the SCPI status reporting structure. See the Index for commands not listed in the following table.

**Table 24: Status commands**

Header	Description
STATus:ACPower:EVENTs?	Returns the current events for the ACPR measurement.
STATus:BIBEmissions:EVENTs?	Returns the current events and status conditions for the Bluetooth InBand Emission measurement.
STATus:BLUEtooth:CONSte:EVENTs?	Returns the current events and status conditions for the Bluetooth Constellation measurement.
STATus:BLUEtooth:EDIAGram:EVENTs?	Returns the current events and status conditions for the Bluetooth Eye Diagram measurement.
STATus:BLUEtooth:FDVTime:EVENTs?	Returns the current events and status conditions for the Frequency Deviation versus Time measurement.
STATus:P25:EDIAGram:EVENTs?	Returns the current events and status conditions for the P25 Eye Diagram measurement.
STATus:{AM FM PM}:EVENTs?	Returns the current events for the AM/FM/PM measurement.
STATus:AUDio:SPECTrum:EVENTs?	Returns the current events and status conditions for the audio measurement.

Table continued...

Header	Description
STATus:AUDio:SUMMery:EVENTs?	Returns the current events and status conditions for the audio measurement.
STATus:AVTime:EVENTs?	Returns the current events for the Amplitude versus Time measurement.
STATus:CCDF:EVENTs?	Returns the current events for the CCDF measurement.
STATus:CONSte[:MEASview<x>]:EVENTs?	Returns the current events and status conditions for the constellation measurement.
STATus:DIQVtime[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the Demod I&Q versus Time measurement.
STATus:DPX:EVENTs?	Returns the current events for the DPX spectrum measurement.
STATus:DPX:EVENTs?	Returns the current events for the DPX spectrum measurement.
STATus:EDlagram[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the eye diagram measurement.
STATus:EVM[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the EVM versus Time measurement.
STATus:FDVTime[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the Frequency deviation versus Time measurement.
STATus:{FSETtling PSETtling}:EVENTs?	Returns the current events for the specified settling time measurement.
STATus:FVTime[:MEASview<y>]:EVENTs?	Returns the current events for the Frequency versus Time measurement.
STATus:IQVTime[:MEASview<y>]:EVENTs?	Returns the current events for the RF I&Q versus Time measurement.
STATus:MCPower:EVENTs?	Returns the current events for the MCPDR measurement.
STATus:MERRor[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the Magnitude error versus Time measurement.
STATus:OBWidth:EVENTs?	Returns the current events for the Occupied Bandwidth measurement.
STATus:OFDM:CONSte:EVENTs?	Returns the current events and status condition for the OFDM Constellation measurement.
STATus:OFDM:EVM:EVENTs?	Returns the current events and status condition for the OFDM EVM (Error Vector Magnitude) versus Time measurement.
STATus:OFDM:MERRor:EVENTs?	Returns the current events and status condition for the OFDM Magnitude versus Time measurement.
STATus:OFDM:PERRor:EVENTs?	Returns the current events and status condition for the OFDM Phase error versus Time measurement.
STATus:OFDM:POWer:EVENTs?	Returns the current events and status condition for the OFDM Power measurement.
STATus:OFDM:STABle:EVENTs?	Returns the current events and status condition for the OFDM Symbol table measurement.
STATus:OPERation:CONDition?	Queries the contents of the OCR.
STATus:OPERation:ENABle	Sets or queries the mask for the OENR.

Table continued...



Header	Description
STATus:OPERation[:EVENT]?	Queries the contents of the OEVR.
STATus:OPERation:NTRansition	Sets or queries the value of the negative transition filter.
STATus:OPERation:PTRansition	Sets or queries the value of the positive transition filter.
STATus:PERRor[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the Phase error versus Time measurement.
STATus:PHVTime[:MEASview<y>]:EVENTs?	Returns the current events for the Phase versus Time measurement.
STATus:PNOise:EVENTs?	Returns the current events for the phase noise measurement.
STATus:PRESet	Presets a status byte.
STATus:PULSe:RESult[:MEASview<y>]:EVENTs?	Returns the current events for the pulse table measurement.
STATus:PULSe:STATistics[:MEASview<y>]:EVENTs?	Returns the current events for the pulse statistics measurement.
STATus:PULSe:TRACe[:MEASview<y>]:EVENTs?	Returns the current events for the pulse trace measurement.
STATus:QUEStionable:CONDition?	Queries the contents of the QCR.
STATus:QUEStionable:ENABle	Sets or queries the mask for the OENR.
STATus:QUEStionable[:EVENT]?	Queries the contents of the QER.
STATus:QUEStionable:NTRansition	Sets or queries the value of the negative transition filter.
STATus:QUEStionable:PTRansition	Sets or queries the value of the positive transition filter.
STATus:QUEStionable:CALibration:CONDition?	Queries the contents of the questionable calibration condition register.
STATus:QUEStionable:CALibration:ENABle	Sets or queries the mask for the questionable calibration enable register.
STATus:QUEStionable:CALibration[:EVENT]?	Queries the contents of the questionable calibration event register.
STATus:QUEStionable:CALibration:NTRansition	Sets or queries the value of the negative transition filter.
STATus:QUEStionable:CALibration:PTRansition	Sets or queries the value of the positive transition filter.
STATus:QUEStionable:FREQuency:CONDition?	Queries the contents of the questionable frequency condition register.
STATus:QUEStionable:FREQuency:ENABle	Sets or queries the mask for the questionable frequency enable register.
STATus:QUEStionable:FREQuency[:EVENT]?	Queries the contents of the questionable frequency event register.
STATus:QUEStionable:FREQuency:NTRansition	Sets or queries the value of the negative transition filter.
STATus:QUEStionable:FREQuency:PTRansition	Sets or queries the value of the positive transition filter.
STATus:SEM:EVENTs?	Returns the current events and status conditions for the Spectral Emissions Mask measurement.
STATus:SGRam[:MEASview<y>]:EVENTs?	Returns the current events for the spectrogram measurement.
STATus:SPEctrum[:MEASview<y>]:EVENTs?	Returns the current events for the spectrum measurement.
STATus:SPURious:EVENTs?	Returns the current events for the spurious measurement.
STATus:SQUality[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the signal quality measurement.

Table continued...

Header	Description
STATus:TDIagram[:MEASview<y>]:EVENTs?	Returns the current events and status conditions for the trellis diagram measurement.
STATus:WLAN:CONStellation:EVENTs?	Returns the current status concatenated with WLAN Constellation-specific event information.
STATus:WLAN:CRESPonse:EVENTs?	Returns the current events and status conditions for the WLAN Channel Response display.
STATus:WLAN:EVM:EVENTs?	Returns the current status concatenated with WLAN EVM-specific event information.
STATus:WLAN:MERRor:EVENTs?	Returns the current status concatenated with WLAN Magnitude Error-specific event information.
STATus:WLAN:PERRor:EVENTs?	Returns status concatenated with WLAN Phase Error-specific event information.
STATus:WLAN:PVTIME:EVENTs?	Returns the current status concatenated with WLAN Power vs. Time-specific event information.
STATus:WLAN:STABLE:EVENTs?	Returns status concatenated with WLAN Symbol Table-specific event information.
STATus:WLAN:SUMMery:EVENTs?	Returns the current events and status conditions for the WLAN summary measurement.
<b>STATus:TXGain subgroup</b>	<b>Transmission Gain measurements</b>
STATus:TXGain:EVENTs?	Queries events for the Transmission Gain display.
<b>STATus:LTE subgroup</b>	<b>LTE analysis</b>
STATus:LTE:ACLR:EVENTs?	Returns the current events and status conditions for this display.
STATus:LTE:CHSPepectrum:EVENTs?	Returns the current events and status conditions for this display.
STATus:LTE:CONStellation:EVENTs?	Returns the current events and status conditions for this display.
STATus:LTE:PVTIME:EVENTs?	Returns the current events and status conditions for this display.

## Source commands

Use the SOURce commands to control output power level of the tracking generator. See the Index for commands not listed in the following table.

These commands only apply when a RSA500A and RSA600A Series instrument with Option 04 (Tracking Generator) installed is connected.

**Table 25: Source commands**

Command	Description
SOURce:TXGain:POWer	Sets or queries the output power level of the tracking generator.

## System commands

Use the SYSTem commands to set or query system parameters for operation. See the Index for commands not listed in the following table.

**Table 26: System commands**

Header	Description
SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS	Sets or queries the GPIB address of the instrument.
SYSTem:PRESet:BLUetooth:STANdard	Sets or queries the standard, setup, Retaining Current Reflevel, and Retaining Center Frequency in the Bluetooth standard preset.
SYSTem:PRESet:P25:STANdard	Restores the analyzer defaults for the P25 Standards preset.
SYSTem:DATE	Sets or queries the current date.
SYSTem:ERRor:ALL?	Queries all the error or event information.
SYSTem:ERRor:CODE:ALL?	Queries all the error or event codes.
SYSTem:ERRor:CODE[:NEXT]?	Queries the latest error or event information.
SYSTem:ERRor:COUNt?	Queries the number of errors or events.
SYSTem:ERRor[:NEXT]?	Queries the latest error or event information.
SYSTem:KLOCK	Enables or disables the local lockout operation.
SYSTem:LINK:SETTings	Sets or queries whether or not global settings linking enabled for the measurements displayed.
SYSTem:LINK:SETTings:SYNC	Sets the synchronize settings referring to source for the measurements displayed.
SYSTem:OPTions?	Queries the optional information.
SYSTem:PRESet	Restores the analyzer to the defaults.
SYSTem:PRESet:APPLication	Restores the analyzer to the defaults for the application preset type.
SYSTem:PRESet:APPLication:ACTion	Sets or queries the preset action for the application preset type.
SYSTem:PRESet:APPLication:SELected	Sets or queries the presets for the application preset type.
SYSTem:PRESet:DPX	Presets the analyzer.
SYSTem:PRESet:DPX:ACTion	Sets or queries the preset action for the DPX preset type.
SYSTem:PRESet:DPX:SELected	Sets or queries the presets for the DPX preset type.
SYSTem:PRESet[:MAIN]	Restores the analyzer to the defaults.
SYSTem:PRESet:MAIN:ACTion	Sets or queries the preset action for the main preset type.
SYSTem:PRESet:MAIN:SELected	Sets or queries the presets for the main preset type.
SYSTem:PRESet:STANdards	Restores the instrument to the defaults for the WLAN standards preset type.
SYSTem:PRESet:STANdards:ACTion	Sets or queries the preset action for the standards preset type.
SYSTem:PRESet:USER	Restores the analyzer to the defaults for the user preset type.
SYSTem:PRESet:USER:ACTion	Sets or queries the preset action for the user preset type.
SYSTem:PRESet:USER:SELected	Sets or queries the presets for the user preset type.
SYSTem:PRESet:WLAN:STANdard	Sets or queries the standard and bandwidth presets for the WLAN standards preset type.

Table continued...

Header	Description
SYSTem:TIME	Sets or queries the current time.
SYSTem:VERSion?	Queries the version of the SCPI.
SYSTem:PRESet:LTE:STANdard	Sets or queries LTE standard preset parameters.

## Trace commands

Use the TRACe commands to select trace type and to control trace arithmetic. See the Index for commands not listed in the following table.

**Table 27: Trace commands**

Commands	Description
<b>TRACe&lt;x&gt;:{AM FM PM} subgroup</b>	<b>AM/FM/PM measurement</b>
TRACe<x>:{AM FM PM}	Determines whether or not to show the specified trace.
TRACe:{AM FM PM};DETection	Sets or queries the display detector, the method to be used for decimating traces to fit the available horizontal space on screen.
TRACe:{AM FM PM};FREeze	Sets or queries whether or not to freeze the trace display.
TRACe:{AM FM PM};FUNction	Sets or queries the trace function.
<b>TRACe&lt;x&gt;:Bluetooth subgroup</b>	<b>Bluetooth measurement</b>
TRACe<x>:BLUEtooth:CONSte	Determines whether or not to show the specified trace.
TRACe<x>:BLUEtooth:CONSte:FREeze	Sets or queries whether or not to freeze the specified trace.
TRACe<x>:BLUEtooth:CONSte:MODE	Sets or queries the trace display mode.
TRACe<x>:BLUEtooth:CONSte:SElect	Selects the specified trace.
TRACe<x>:BLUEtooth:EDIagram:ENABle:I	Determines whether to show or hide the I trace.
TRACe<x>:BLUEtooth:EDIagram:ENABle:Q	Determines whether to show or hide the Q trace.
TRACe<x>:BLUEtooth:EDIagram:SElect:I	Selects the I trace or queries if the I trace is selected or not.
TRACe<x>:BLUEtooth:EDIagram:SElect:Q	Selects the Q trace or queries if the Q trace is selected or not.
TRACe<x>:BLUEtooth:FDVTime:MODE	Sets or queries whether to display the Frequency Deviation vs. Time trace as vectors or symbols (points).
<b>TRACe:P25 subgroup</b>	<b>APCO P25 measurement</b>
TRACe<x>:P25:CONStellation	Determines whether or not to show the specified trace.
TRACe<x>:P25:CONStellation:FREeze	Determines whether or not to freeze (halt updates of) the specified trace.
TRACe<x>:P25:CONStellation:MODE	Sets or queries how to display the trace content.
TRACe<x>:P25:CONStellation:SElect	Selects the specified trace.
TRACe<x>:P25:EDIagram:ENABle:I	Determines whether to show or hide the I trace.
TRACe<x>:P25:EDIagram:ENABle:Q	Determines whether to show or hide the Q trace.
TRACe<x>:P25:EDIagram:SElect:I	Selects the I trace or queries if the I trace is selected or not.
TRACe<x>:P25:EDIagram:SElect:Q	Selects the Q trace or queries if the Q trace is selected or not.

Table continued...

Commands	Description
<b>TRACe:Noise subgroup</b>	<b>Noise Figure and Gain measurement</b>
TRACe<x>:NOISe:FIGure:AVERageCOUNT	Sets or queries the average counts for the specified trace.
TRACe<x>:NOISe:FIGure:COUNT:ENABLE	Sets to enable the count feature, or queries the enables status, of the specified trace.
TRACe<x>:NOISe:FIGure:DETection	Sets or queries the trace detection method.
TRACe<x>:NOISe:FIGure:FREeze	Determines whether or not to freeze (halt updates of) the specified trace.
TRACe<x>:NOISe:Figure:FUNCTion	Sets or queries the trace processing method (function) in the display.
TRACe<x>:NOISe:FIGure:SElect	Selects the specified trace.
TRACe<x>:NOISe:FIGure:SHOW	Determines whether or not to show the specified trace.
TRACe<x>:NOISe:GAIN:AVERage:COUNT	Sets or queries the average counts for the specified trace.
TRACe<x>:NOISe:GAIN:COUNT:ENABLE	Enables or disables the count feature for the specified trace.
TRACe<x>:NOISe:GAIN:DETection	Sets or queries the trace detection method.
TRACe<x>:NOISe:GAIN:FREeze	Determines whether or not to freeze (halt updates of) the specified trace.
TRACe<x>:NOISe:GAIN:FUNCTion	Sets or queries the trace processing method (function) in the display.
TRACe<x>:NOISe:GAIN:SElect	Selects the specified trace.
TRACe<x>:NOISe:GAIN:SHOW	Determines whether or not to show the specified trace.
TRACe<x>:NOISe:TEMPerature:AVERageCOUNT	Sets or queries the average counts for the specified trace.
TRACe<x>:NOISe:TEMPerature:COUNT:ENABLE	Enables or disables the count feature for the specified trace.
TRACe<x>:NOISe:TEMPerature:DETection	Sets or queries the trace detection method.
TRACe<x>:NOISe:TEMPerature:FREeze	Determines whether or not to freeze (halt updates of) the specified trace.
TRACe<x>:NOISe:TEMPerature:FUNCTion	Sets or queries the trace processing method (function) in the display.
TRACe<x>:NOISe:TEMPerature:SElect	Selects the specified trace.
TRACe<x>:NOISe:TEMPerature:SHOW	Determines whether or not to show the specified trace.
TRACe<x>:NOISe:YFACtor:AVERage:COUNT	Sets or queries the average counts for the specified trace.
TRACe<x>:NOISe:YFACtor:COUNT:ENABLE	Enables or disables the count feature for the specified trace.
TRACe<x>:NOISe:YFACtor:DETection	Sets or queries the trace detection method.
TRACe<x>:NOISe:YFACtor:FREeze	Determines whether or not to freeze (halt updates of) the specified trace.
TRACe<x>:NOISe:YFACtor:FUNCTion	Sets or queries the trace processing method (function) in the display.
TRACe<x>:NOISe:YFACtor:SElect	Selects the specified trace.
TRACe<x>:NOISe:YFACtor:SHOW	Determines whether or not to show the specified trace.
<b>TRACe&lt;x&gt;:AVTime subgroup</b>	<b>Amplitude versus time measurement</b>

Table continued...

Commands	Description
TRACe<x>:AVTime[:MEASview<y>]	Sets or queries whether or not to show the specified trace.
TRACe:{FSETtling PSETtling}:AVERAge:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe<x>:AVTime[:MEASview<y>]:AVERAge:COUNT	Sets or queries the number of traces to combine.
TRACe<x>:AVTime[:MEASview<y>]:AVERAge:RESet	Clears the average data and resets the average counter.
TRACe<x>:AVTime[:MEASview<y>]:COUNT	Sets or queries the count for the Max or Min Hold trace.
TRACe<x>:AVTime[:MEASview<y>]:COUNT:ENABle	Sets or queries whether or not to enable the count for Max/Min Hold.
TRACe<x>:AVTime[:MEASview<y>]:COUNT:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:AVTime[:MEASview<y>]:DETEction	Sets or queries the display detector.
TRACe<x>:AVTime[:MEASview<y>]:FREeze	Sets or queries whether or not to freeze the display of the trace.
TRACe<x>:AVTime[:MEASview<y>]:FUNction	Sets or queries the trace function.
TRACe<x>:AVTime[:MEASview<y>]:LEFToperand	Sets or queries the left operand for the math trace.
TRACe<x>:AVTime[:MEASview<y>]:RIGHTooperand	Sets or queries the right operand for the math trace.
TRACe<x>:AVTime[:MEASview<y>]:SELEct	Sets or queries the trace number to display the readout.
<b>TRACe&lt;x&gt;:CCDF subgroup</b>	<b>CCDF measurement</b>
TRACe<x>:CCDF:FREeze	Sets or queries whether or not to freeze the display of the trace.
TRACe<x>:CCDF:SELEct	Sets or queries the trace number to display the readout.
TRACe<x>:CCDF:SHOW	Sets or queries whether to show or hide the trace.
TRACe<x>:CCDF:X	Sets or queries the horizontal position of the measurement pointer.
TRACe<x>:CCDF:Y?	Queries the vertical position (CCDF value) of the measurement pointer.
<b>TRACe:CONStellation subgroup (Option 21 only)</b>	<b>Constellation measurement</b>
TRACe<x>:CONStellation[:MEASview<y>]	Specifies whether or not to show the specified trace (Trace 1 or 2) in the Constellation display.
TRACe<x>:CONStellation[:MEASview<y>]:FREeze	Halts updates to the specified trace in the Constellation display.
TRACe<x>:CONStellation[:MEASview<y>]:MODE	Sets or queries how to display the constellation trace.
TRACe<x>:CONStellation[:MEASview<y>]:Q:OFFSet	Sets the Q offset in the Constellation display to Use shared Pref, remove Q offset, or include Q offset.
TRACe<x>:CONStellation[:MEASview<y>]:SELEct	Sets or queries the trace to which the Show, Freeze, Content, and Q Offset settings apply.
<b>TRACe:DIQVtime subgroup (Option 21 only)</b>	<b>Demodulated I&amp;Q versus time measurement</b>
TRACe:DIQVtime[:MEASview<y>]:ENABle:I	Enable or disables the trace I.
TRACe:DIQVtime[:MEASview<y>]:ENABle:Q	Enable or disables the Q trace.
TRACe:DIQVtime[:MEASview<y>]:MODE	Sets or queries whether to display the Demod I&Q vs Time trace as vectors or symbols.
TRACe:DIQVtime[:MEASview<y>]:SELEct:I	Selects the I trace.
TRACe:DIQVtime[:MEASview<y>]:SELEct:Q	Selects the Q trace.
<b>TRACe&lt;x&gt;:DPX subgroup</b>	<b>DPX spectrum measurement</b>

Table continued...

Commands	Description
TRACe<x>:DPX	Sets or queries whether or not to show the waveform.
TRACe<x>:DPX:AVERAge:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe<x>:DPX:COLor:CURVe	Sets or queries the color intensity.
TRACe<x>:DPX:COLor:INTensity	Sets or queries the color intensity in the DPX spectrum view.
TRACe<x>:DPX:COLor:SCALe:AUTO	Automatically adjusts the Max and Min color settings to display the broadest range of colors.
TRACe<x>:DPX:DETection	Sets or queries the function.
TRACe<x>:DPX:DGRAM:SElect:LINE	Sets or queries a line in the DPXogram display to send to the DPX spectrum display.
TRACe<x>:DPX:DOT:PERsistent	Sets or queries whether to enable or disable the dot persistence.
TRACe<x>:DPX:DOT:PERsistent:TYPE	Sets or queries the persistence type.
TRACe<x>:DPX:DOT:PERsistent:VARiable	Sets or queries the length of time that data points are displayed.
TRACe<x>:DPX:FReeze	Sets or queries whether or not to freeze the display of the trace.
TRACe<x>:DPX:FUNCTion	Sets or queries the trace function for the in the DPX spectrum view.
TRACe<x>:DPX:LEFToperand	Sets or queries the left operand for the math trace.
TRACe<x>:DPX:RIGHToperand	Sets or queries the right operand for the math trace.
TRACe<x>:DPX:SElect	Sets or queries the trace number to display the readout.
<b>TRACe:EDiagram subgroup (Option 21 only)</b>	<b>Eye diagram</b>
TRACe:EDiagram[:MEASview<y>]:ENABLe:I	Determines whether to show or hide the I trace in the eye diagram.
TRACe:EDiagram[:MEASview<y>]:ENABLe:Q	Determines whether to show or hide the Q trace in the eye diagram.
TRACe:EDiagram[:MEASview<y>]:SElect:I	Selects the I trace in the eye diagram. The query version of this command returns whether the I trace is selected or not.
TRACe:EDiagram[:MEASview<y>]:SElect:Q	Selects the Q trace in the eye diagram. The query version of this command returns whether the Q trace is selected or not.
<b>TRACe:EVM subgroup</b>	<b>EVM measurement</b>
TRACe:EVM[:MEASview<y>]:MODE	Sets or queries whether to display the EVM versus Time trace as vectors or symbols (points).
<b>TRACe:FDVTime subgroup</b>	<b>Frequency deviation versus time measurement</b>
TRACe:FDVTime:MODE	Sets or queries whether to display the Frequency Deviation versus Time trace as vectors or symbols.
<b>TRACe:FVTime subgroup</b>	<b>Frequency versus time measurement</b>
TRACe:FVTime[:MEASview<y>]	Sets or queries whether or not to show the trace.
TRACe:FVTime[:MEASview<y>]:AVERAge:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe:FVTime[:MEASview<y>]:COUNT	Sets or queries the count for the Max or Min Hold trace.
TRACe:FVTime[:MEASview<y>]:COUNT:ENABLe	Sets or queries whether or not to enable the count for Max or Min Hold.
TRACe:FVTime[:MEASview<y>]:COUNT:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:FVTime[:MEASview<y>]:DETection	Enables or queries the type of detection for the specified trace.

Table continued...

Commands	Description
TRACe:FVTime[:MEASview<y>]:FREeze	Sets or queries whether or not to freeze the display of the trace.
TRACe:FVTime[:MEASview<y>]:FUNction	Sets or queries the trace function.
<b>TRACe:IQVTime subgroup</b>	<b>RF I&amp;Q versus time measurement</b>
TRACe:IQVTime[:MEASview<y>]:AVERage:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe:IQVTime[:MEASview<y>]:COUNT	Sets or queries the count for the Max or Min Hold trace.
TRACe:IQVTime[:MEASview<y>]:COUNT:ENABle	Determines whether to enable or disable the count for the Max or Min Hold trace.
TRACe:IQVTime[:MEASview<y>]:DETEction	Sets or queries whether or not to enable the count for Max/Min Hold.
TRACe:IQVTime[:MEASview<y>]:COUNT:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe:IQVTime[:MEASview<y>]:ENABle:I	Sets or queries whether to show or hide the trace I.
TRACe:IQVTime[:MEASview<y>]:ENABle:Q	Sets or queries whether to show or hide the trace Q.
TRACe:IQVTime[:MEASview<y>]:FREeze	Sets or queries whether to freeze the IQ traces.
TRACe:IQVTime[:MEASview<y>]:FUNction	Sets or queries the trace function.
TRACe:IQVTime[:MEASview<y>]:SELect:I	Sets or queries whether to choose the I trace.
TRACe:IQVTime[:MEASview<y>]:SELect:Q	Sets or queries whether to choose the Q trace.
<b>TRACe:MERRor subgroup</b>	<b>Magnitude Error measurement</b>
TRACe:MERRor[:MEASview<y>]:MODE	Sets or queries whether to display the Magnitude error trace as vectors or symbols.
<b>TRACe:OBWidth subgroup</b>	<b>Occupied Bandwidth measurement</b>
TRACe:OBW:MAXHold	Determines whether to enable or disable the Max Hold trace.
<b>TRACe:OFDM subgroup</b>	<b>OFDM measurements</b>
TRACe:OFDM:CONSte	Determines whether or not to show the trace.
TRACe:OFDM:CONSte:FREeze	Determines whether or not to freeze the display of the trace.
TRACe<x>:OFDM:EVM	Determines whether or not to show the specified trace.
TRACe<x>:OFDM:EVM:FREeze	Determines whether or not to freeze the display of the specified trace.
TRACe<x>:OFDM:EVM:SELect	Sets or queries the specified trace to display.
TRACe<x>:OFDM:MERRor	Determines whether or not to show the specified trace.
TRACe<x>:OFDM:MERRor:FREeze	Determines whether or not to freeze the display of the specified trace.
TRACe<x>:OFDM:MERRor:SELect	Sets or queries the specified trace to display.
TRACe<x>:OFDM:PERRor	Determines whether or not to show the specified trace.
TRACe<x>:OFDM:PERRor:FREeze	Determines whether or not to freeze the display of the specified trace.
TRACe<x>:OFDM:PERRor:SELect	Sets or queries the specified trace to display.
TRACe<x>:OFDM:POWER	Determines whether or not to show the specified trace.
Table continued...	



Commands	Description
TRACe<x>:OFDM:POWer:FREeze	Determines whether or not to freeze the display of the specified trace.
TRACe<x>:OFDM:POWer:SElect	Sets or queries the specified trace to display.
TRACe:OFDM:FLATness:FREeze	Sets or queries whether to freeze the average trace
TRACe:OFDM:FLATness:SHOW	Displays or hides the trace on the OFDM Spectral Flatness display.
<b>TRACe:PERRpr subgroup</b>	<b>Phase Error measurement</b>
TRACe:PERRor[:MEASview<y>]:MODE	Sets or queries whether to display the Frequency deviation vs Time trace as vectors or symbols.
<b>TRACe:PHVTime subgroup</b>	<b>Phase versus time measurement</b>
TRACe:PHVTime[:MEASview<y>]	Sets or queries whether or not to show the trace.
TRACe:PHVTime[:MEASview<y>]:AVERage:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe:PHVTime[:MEASview<y>]:COUNT	Sets or queries the count for the Max or Min Hold trace.
TRACe:PHVTime[:MEASview<y>]:COUNT:ENABle	Sets or queries whether or not to enable the count for Max/Min Hold.
TRACe:PHVTime[:MEASview<y>]:COUNT:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:PHVTime[:MEASview<y>]:DETEction	Sets or queries the display detector.
TRACe<x>:{FSETtling PSETtling}:FREeze	Sets or queries whether to freeze the trace.
TRACe:PHVTime[:MEASview<y>]:FREeze	Determines whether to freeze the trace display in the Phase versus Time measurement.
TRACe:PHVTime[:MEASview<y>]:FUNCTion	Sets or queries the trace function.
<b>TRACe:PNOise subgroup (Option 11 only)</b>	<b>Phase noise measurement</b>
TRACe<x>:{FSETtling PSETtling}:SElect	Selects a trace or queries the currently selected trace.
TRACe<x>:{FSETtling PSETtling}:SHOW	Sets or queries whether to show or hide the trace.
TRACe:{FSETtling PSETtling}:AVERage:ENABle	Sets or queries the number of data points for smoothing the trace.
TRACe<x>:PNOise:SElect	Selects the trace in the phase noise measurement.
TRACe<x>:PNOise:SHOW	Shows or hides the specified trace in the phase noise view.
TRACe<x>:PNOise:SMOothing:COUNT	Sets or queries the number of data points to take the moving average for smoothing the trace.
TRACe<x>:PNOise:SMOothing:ENABle	Sets or queries whether to enable smoothing the specified trace.
TRACe<x>:PNOise:SMOothing:RESet	Restarts the smoothing process.
<b>TRACe:{FSETtling PSETtling} subgroup (Option 12 only)</b>	<b>Settling time measurements</b>
TRACe:{FSETtling PSETtling}:AVERage:COUNT	Sets or queries the number of traces to combine.
TRACe:{FSETtling PSETtling}:AVERage:COUNT:CURRent?	Queries the current running average count value.
TRACe:{FSETtling PSETtling}:AVERage:ENABle	Determines whether to enable or disable averaging the trace(s).
TRACe:{FSETtling PSETtling}:RESet	If trace averaging is enabled, this command resets the current averaged trace to 0.
TRACe:{FSETtling PSETtling}:SMOothing:COUNT	Sets or queries the number of data points to take the moving average for smoothing the traces.

Table continued...

Commands	Description
TRACe:{FSETtling PSETtling}:SMOothing:ENABLE	Determines whether to enable or disable smoothing the trace(s).
TRACe<x>:{FSETtling PSETtling}:FREeze	Determines whether to freeze the specified trace in the Settling Time measurement display.
TRACe<x>:{FSETtling PSETtling}:SElect	Selects the trace in the settling time measurement.
TRACe<x>:{FSETtling PSETtling}:SHOW	Determines whether to show or hide the specified trace in the settling time view.
<b>TRACe:SEM subgroup</b>	<b>Spectral Emissions Mask measurements</b>
TRACe:SEM:COUNT	Sets or queries how many acquisitions run in the single acquisition mode for multi-trace functions.
TRACe:SEM:COUNT:RESet	Clears the multi-function data and counter and restarts the process.
TRACe:SEM:FUNction	Sets or queries the trace function.
<b>TRACe:SGRam subgroup</b>	<b>Spectrogram measurement</b>
TRACe:SGRam[:MEAvieW<y>]:DETection	Sets or queries the method to be used for decimating traces to fit the on screen.
TRACe:SGRam[:MEAvieW<y>]:FREeze	Sets or queries whether or not to freeze the spectrogram display.
TRACe:SGRam[:MEAvieW<y>]:FUNction	Sets or queries the trace function for the specified trace in the spectrogram.
TRACe:SGRam[:MEAvieW<y>]:FUNction:TIME	Sets or queries the number of traces to combine for the trace function.
TRACe:SGRam[:MEAvieW<y>]:SElect:LINE	Sets or queries the line number to send to the spectrum display.
<b>TRACe&lt;x&gt;:SPEctrum subgroup</b>	<b>Spectrum measurement</b>
TRACe<x>:SPEctrum[:MEAvieW<y>]	Sets or queries whether to show or hide the specified trace.
TRACe<x>:SPEctrum[:MEAvieW<y>]:AVERage:COUNT	Sets or queries the number of traces to combine for averaging.
TRACe<x>:SPEctrum[:MEAvieW<y>]:AVERage:RESet	Clears the average data and resets the average counter.
TRACe<x>:SPEctrum[:MEAvieW<y>]:COUNT	Sets or queries the count for the Max or Min Hold trace.
TRACe<x>:SPEctrum[:MEAvieW<y>]:COUNT:ENABLE	Sets or queries whether or not to enable the count for Max/Min Hold.
TRACe<x>:SPEctrum[:MEAvieW<y>]:COUNT:RESet	Clears the Max or Min Hold data and counter, and restarts the process.
TRACe<x>:SPEctrum[:MEAvieW<y>]:DETection	Sets or queries the display detector.
TRACe<x>:SPEctrum[:MEAvieW<y>]:FREeze	Sets or queries whether or not to freeze the display of the trace.
TRACe<x>:SPEctrum[:MEAvieW<y>]:FUNction	Sets or queries the trace function.
TRACe<x>:SPEctrum[:MEAvieW<y>]:LEFToperand	Sets or queries the left operand for the math trace.
TRACe<x>:SPEctrum[:MEAvieW<y>]:RIGHToperand	Sets or queries the right operand for the math trace.
TRACe<x>:SPEctrum[:MEAvieW<y>]:SElect	Sets or queries the trace number to display the readout.
<b>TRACe:SPURious subgroup</b>	<b>Spurious measurement</b>
TRACe:SPURious:COUNT	Sets or queries the count for the Max Hold or Average trace.
TRACe:SPURious:COUNT:ENABLE	Sets or queries whether to enable the count for Max Hold or Average.

Table continued...

Commands	Description
TRACe:SPURious:COUNT:RESet	Clears Max Hold or Average data and counter, and restarts the process.
TRACe:SPURious:FREeze	Determines whether or not to freeze the display of the trace.
TRACe:SPURious:FUNCTion	Sets or queries the trace function.
<b>TRACe:TOVerview subgroup</b>	<b>All measurements</b>
TRACe1:TOVerview[:MEASview<y>]	Enables display of or queries the display status of the specified trace.
TRACe1:TOVerview[:MEASview<y>]:AVERAge:COUNT	Sets or queries the number of traces averaged to generate the specified trace.
TRACe1:TOVerview[:MEASview<y>]:COUNT	Enables or queries the count set for the specified trace.
TRACe1:TOVerview[:MEASview<y>]:COUNT:ENABLE	Enables or queries the Average count for the specified trace.
TRACe1:TOVerview[:MEASview<y>]:COUNT:RESet	Sets or queries whether or not to freeze the display of the trace.
TRACe1:TOVerview[:MEASview<y>]:DETection	Enables or queries the type of detection for the specified trace.
TRACe1:TOVerview[:MEASview<y>]:FREeze	Enables or queries a halt to acquisition updates for the specified trace.
TRACe1:TOVerview[:MEASview<y>]:FUNCTion	Enables or queries the selected Function for the specified trace.
<b>TRACe:WLAN group</b>	<b>Wireless LAN measurements</b>
TRACe:WLAN:FLATness:FREeze	Sets or queries whether to freeze the average trace on the WLAN Spectral Flatness display.
TRACe:WLAN:FLATness:SHOW	Displays or hides the trace on the WLAN Spectral Flatness display.
TRACe<x>:WLAN:CONSte	Turns the selected trace on or off in the Constellation plot display.
TRACe<x>:WLAN:CONSte:FREeze	Turns the freeze state of the selected trace on or off.
TRACe<x>:WLAN:CRESPonse:SElect	Sets or queries the active trace in the WLAN Channel Response.
TRACe<x>:WLAN:EVM:FREeze	Turns the freeze state of the selected trace on or off.
TRACe<x>:WLAN:EVM[:SHOW]	Displays or hides the selected trace.
TRACe<x>:WLAN:MERRor:FREeze	Turns the freeze state of the selected trace on or off.
TRACe<x>:WLAN:MERRor[:SHOW]	Displays or hides the selected trace.
TRACe<x>:WLAN:PERRor:FREeze	Turns the freeze state of the selected trace on or off.
TRACe<x>:WLAN:PERRor[:SHOW]	Displays or hides the selected trace.
<b>TRACe:LTE subgroup</b>	
TRACe<x>:LTE:ACLR:COUNT	Sets or queries how many acquisitions run in the single acquisition mode for multitrace functions.
TRACe<x>:LTE:ACLR:COUNT:RESet	Clears the multifunction (Max Hold or Average) data and counter and restarts the process.
TRACe<x>:LTE:ACLR:FUNCTion	Sets or queries the trace function.
TRACe<x>:LTE:CONSte:SYNChronization: SEQUence:STATe?	Returns whether or not a measurement signal (Primary Synchronization Signal/Secondary Synchronization Signal) is selected for the specified trace.

## Trace mnemonics

### Trace Mnemonics

Multiple traces can be used in some measurement displays. The traces are specified by the trace specifier TRACe<x> (<x>=1 to 5) which is defined for each measurement display as follows.

**Table 28: Trace mnemonics**

Measurement display	TRACe1	TRACe2	TRACe3	TRACe4	TRACe5
Amplitude versus Time	Trace 1	Trace 2	Math trace	NA	NA
Phase noise	Trace 1	Trace 2	NA	NA	NA
Spectrum	Trace 1	Trace 2	Trace 3	Math trace	Spectrogram



**Note:** Valid traces depend on commands. Refer to each command description.

## Trigger commands

Use the TRIGger commands to set up the trigger system. See the Index for commands not listed in the following table.

**Table 29: Trigger commands**

Commands	Description
*TRG (No Query Form)	Generates a trigger. It produces the same effect as the Force Trigger button on the Trigger control panel. This command is valid when the trigger mode is Triggered.
TRIGger:ADVanced:SWEep:MODE	Sets or enables the analyzer to wait for a trigger for each acquisition in a swept spectrum or other measurement requiring multiple acquisitions worth of span.
TRIGger:DPX:SHOW:FRAMES	Sets or queries whether to trigger each segment in the swept mode.
TRIGger:DPX:TDM:SHOW:ALL	Sets or queries the selections of the DPX Prefs control panel tab page.
TRIGger:EVENT:EXternal:SLOPe	Sets or queries the signal slope of the External input to determine a trigger event.
TRIGger:EVENT:INPut:LEVel	Sets or queries the power level that the signal must pass through to cause a trigger event when the trigger type is set to Power.
TRIGger:EVENT:INPut:SLOPe	Sets or queries the signal slope of the RF Input to determine a trigger event.
TRIGger:EVENT:INPut:TYPE	Sets or queries the type of trigger. (Power is the only available setting.)
TRIGger:EVENT:SOURce	Sets or queries the source of the signal to be monitored for a trigger event.
TRIGger:MASK:NEW	Loads a new frequency mask.

Table continued...

Commands	Description
TRIGger:MASK:NEW:AUTO	Draws a new frequency mask automatically.
TRIGger:MASK:OPEN	Opens a trigger mask with a specified file.
TRIGger:MASK:SAVE	Saves the current trigger mask to a specified file.
TRIGger:SAVE:COUNT:MAXimum	Sets or queries the number of times a Save on Trigger action stores a file.
TRIGger:SAVE:COUNT[:STATe]	Sets or queries the Maximum total number of files feature.
TRIGger:SAVE:DATA:FORMat	Sets or queries the file format used when saving the acquisition data on a trigger event.
TRIGger:SAVE:DATA[:STATe]	Sets or queries to save data when a trigger event occurs.
TRIGger:SAVE:PICTure:FORMat	Sets or queries the file format used when saving a picture of the display when a trigger event occurs.
TRIGger:SAVE:PICTure[:STATe]	Sets or queries the save picture on trigger setting.
TRIGger[:SEquence]:ADVanced:HOLDoff	FIRST Sets or queries the trigger holdoff time.
TRIGger[:SEquence]:ADVanced:HOLDoff:ENABle	Sets or queries the trigger delay time.
TRIGger[:SEquence]:ADVanced:SWEep:MODE	Determines whether or not to trigger each segment in the swept acquisition mode.
TRIGger[:SEquence]:ADVanced:SWEpt:SEGment: ENABle	Determines whether or not to wait for a trigger for each acquisition in a swept spectrum.
TRIGger[:SEquence]:EVENT:A:EDGE:COUPling	Sets or queries the choice of Trigger coupling type.
TRIGger[:SEquence]:EVENT:A:EDGE:HOLDoff:MODE	Sets or queries the trigger hold off mode.
TRIGger[:SEquence]:EVENT:A:EDGE:HOLDoff:TIME	Sets or queries the hold off time.
TRIGger[:SEquence]:EVENT:A:EDGE:LEVEl	Sets or queries where the trigger point occurs for trigger source type.
TRIGger[:SEquence]:EVENT:EXTFront:IMPedance	Sets or queries the impedance of the external trigger input (front).
TRIGger[:SEquence]:EVENT:EXTFront:LEVEl	Sets or queries the trigger level at the external trigger input (front).
TRIGger[:SEquence]:EVENT:EXTFront:SLOPe	Sets or queries the trigger slope of the external trigger input (front).
TRIGger[:SEquence]:EVENT:EXTRear:SLOPe	Sets or queries the trigger slope of the external trigger input (rear).
TRIGger[:SEquence]:EVENT:GATed	Sets or queries the logic for the gated trigger input.
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude	Sets or queries the center vertical level for the density trigger region.
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance	Sets or queries the trigger level for the RF input level trigger.
TRIGger[:SEquence]:EVENT:INPut:DDENsity: FREQuency	Sets or queries the center horizontal level for the density trigger region.
TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance	Sets or queries the left and right extensions of the density trigger region from the center horizontal level.
TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold	Sets or queries the DPX Density threshold that defines a trigger event.
TRIGger[:SEquence]:EVENT:INPut:DDENsity:VIOLation	Sets or queries whether a density value higher or lower than the THReshold value defines a trigger event.

Table continued...

Commands	Description
TRIGger[:SEquence]:EVENT:INPut:FEDGe:LEVel	Sets or queries the trigger level at the frequency edge input.
TRIGger[:SEquence]:EVENT:INPut:FEDGe:LEVel: ACTual?	Queries the actual length of the frequency edge input for the DPX measurement.
TRIGger[:SEquence]:EVENT:INPut:FEDGe:SLOPe	Sets or queries the trigger slope of the frequency edge input.
TRIGger[:SEquence]:EVENT:INPut:FEDGe:THReshold: LEVel	Set or queries the threshold level in the DPX measurement.
TRIGger[:SEquence]:EVENT:INPut:FEDGe:THReshold: STATe	Set or queries the threshold state in the DPX measurement.
TRIGger[:SEquence]:EVENT:INPut:FMASk:{BANDwidth BWiDth}[:RESolution]	Sets or queries the Resolution Bandwidth value to be used in the spectrum view for the frequency mask trigger.
TRIGger[:SEquence]:EVENT:INPut:FMASk:BANDwidth BWiDth[:RESolution]:ACTual?	Sets or queries when the trigger occurs in the frequency mask trigger.
TRIGger[:SEquence]:EVENT:INPut:FMASk:BANDwidth BWiDth[:RESolution]:AUTO	Sets or queries whether to automatically set the Resolution Bandwidth value used in the spectrum view for the frequency mask trigger.
TRIGger[:SEquence]:EVENT:INPut:FMASk:VIOLation	Sets or queries when the analyzer triggers in the frequency mask trigger.
TRIGger[:SEquence]:EVENT:INPut:LEVel	Sets or queries the trigger level for the RF input level trigger.
TRIGger[:SEquence]:EVENT:INPut:RUNT:PULSe	Sets or queries the Runt trigger for a positive or negative going pulse.
TRIGger[:SEquence]:EVENT:INPut:RUNT:PULSe:HIGH: LEVel	Sets or queries the trigger slope for the RF input level trigger.
TRIGger[:SEquence]:EVENT:INPut:RUNT:PULSe:LOW: LEVel	Sets or queries the lower level (second threshold) to qualify a Runt trigger.
TRIGger[:SEquence]:EVENT:INPut:SLOPe	Sets or queries the trigger slope for the RF input level trigger.
TRIGger[:SEquence]:EVENT:INPut:TDBWidth	Sets or queries the time-domain bandwidth.
TRIGger[:SEquence]:EVENT:INPut:TDBWidth:ACTual?	Queries the actual time-domain bandwidth.
TRIGger[:SEquence]:EVENT:INPut:TDBWidth:STATe	Determines whether to set the time-domain bandwidth automatically.
TRIGger[:SEquence]:EVENT:INPut:TYPE	Sets or queries the trigger type for the source of RF input.
TRIGger[:SEquence]:EVENT:SOURce	Sets or queries the trigger event source.
TRIGger[:SEquence]:FORCed	Sets or queries whether to causes a manual trigger.
TRIGger:SEquence:FORCed (No Query Form)	Causes a trigger event to occur when the trigger mode is set to Triggered, regardless of the set trigger conditions.
TRIGger[:SEquence]:IMMediate	Causes a trigger immediately.
TRIGger[:SEquence]:STATus	Sets or queries the trigger mode (Free Run or Triggered).
TRIGger[:SEquence][:STATus]	Sets or queries the trigger mode setting.
TRIGger[:SEquence]:TIME:DELay	Sets or queries the trigger delay time.
TRIGger[:SEquence]:TIME:POSition	Sets or queries the trigger position.
TRIGger[:Sequence]:TIME:POSition:AUTO	Sets or queries the position of the trigger event (automatic or manual) within the acquisition record.
TRIGger[:SEquence]:TIME:QUALified	LAST Sets or queries the timing qualification setting for triggers.
TRIGger[:SEquence]:TIME:QUALified:TIME<x>	Sets or queries the trigger mode (Free Run or Triggered).

Table continued...

Commands	Description
TRIGger[:SEquence]:EVENT:A:EDGE:SETTings	Sets or queries the settings of Trigger source type.
TRIGger[:SEquence]:EVENT:A:EDGE:SLOPe	Sets or queries whether instrument finds the trigger point on rising or falling edge or either edge of a signal.
TRIGger[:SEquence]:EVENT:A:EDGE:SOURce	Sets or queries the channel on which signal is to be monitored for a trigger event.
TRIGger[:SEquence]:EVENT:A:TYPE	Sets or queries the trigger type {supporting only Edge trigger}.
TRIGger[:SEquence]:MODE	Sets or queries the trigger mode setting.
TRIGger:TIME:POSition	Sets or queries the point (in percent) along the acquisition record the trigger event should occur.
TRIGger:TIME:POSition:AUTO	Sets or queries the instrument to automatically set the position of the trigger event.

## Unit commands

Specify fundamental units for measurement.

**Table 30: Unit commands**

Header	Description
UNIT:POWer	Sets or queries the unit of power.

## Retrieving response message

When receiving a query command from the external controller, the analyzer puts the response message on the Output Queue. This message cannot be retrieved unless you perform retrieval operations through the external controller. (For example, call the IBRD subroutine included in the GPIB software of National Instruments.)

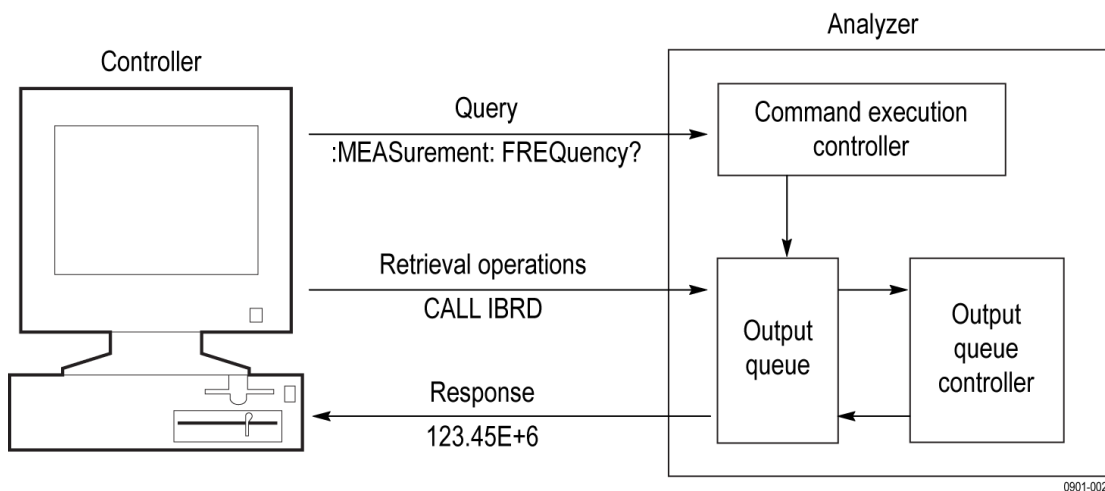


Figure 8: Retrieving response message

When the Output Queue contains a response message, sending another command from the external controller before retrieving this message deletes it from the queue. The Output Queue always contains the response message to the most recent query command.

You can use the MAV bit of the Status Byte Register (SBR) to check whether the Output Queue contains a response message. For details, refer to *Status Byte Register (SBR)*.



---

# Command descriptions

## ABORt (No query form)

Resets the trigger system and places all trigger sequences in the idle state. Any actions related to the trigger system that are in progress, such as a sweep or acquiring a measurement is also aborted.

To start data acquisition, use the INITiate commands.

### Conditions

Measurement views: All

### Group

Abort commands

### Syntax

ABORt

### Related Commands

INITiate:CONTinuous, INITiate:RESume

### Arguments

None

### Examples

ABORt resets the trigger system and stops data acquisition.

## \*CAL (Query only)

### Conditions

Measurement views: All

### Group

IEEE common commands

### Syntax

\*CAL

### Returns

<NR1>=1 indicates that the alignment was successful.

<NR1>=0 indicates that the alignment was unsuccessful.

## Examples

\*CAL performs an internal self-alignment and will return 1 if the alignment is successful.

## Calculate commands

### **CALCulate[:SEARch]:TEST]:LIMit[:STATe]:MATCh:FILE:LOCation**

Sets or queries the destination folder for “search match” output files.

#### Conditions

Measurement views: All

#### Group

Calculate commands

#### Syntax

```
CALCulate[:SEARch|:TEST]:LIMit[:STATe]:MATCh:FILE:LOCation <string>
```

```
CALCulate[:SEARch|:TEST]:LIMit[:STATe]:MATCh:FILE:LOCation?
```

#### Arguments

<string> specifies the destination folder.

#### Returns

<string>

#### Examples

CALCULATE[:SEARCH|:TEST]:LIMIT[:STATE]:MATCH:FILE:LOCATION? might return `c:\rsamap` files, indicating the destination folder.

### **CALCulate[:SEARch]:TEST]:LIMit[:STATe]:MATCh:FILE:NAME:BASE**

Sets or queries the file basename for “search match” output files.

#### Conditions

Measurement views: All

#### Group

Calculate commands

#### Syntax

```
CALCulate[:SEARch|:TEST]:LIMit[:STATe]:MATCh:FILE:NAME:BASE<string>
```

```
CALCulate[:SEARCH|:TEST]:LIMIT[:STATE]:MATCH:FILE:NAME:BASE?
```

## Arguments

<string> specifies the output file.

## Returns

<string>

## Examples

```
CALCULATE[:SEARCH|:TEST]:LIMIT[:STATE]:MATCH:FILE:NAME:BASEBASE sets the base filename as BASE
```

## CALCulate:ACPower:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:DELTA:X?
```

## Related Commands

CALCulate:MARKer:ADD, CALCulate:ACPower:MARKer<x>:DELTA:Y?

## Returns

<NRf> Delta marker frequency for the selected marker.

## Examples

```
CALCULATE:ACPOWER:MARKER1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.
```

## CALCulate:ACPower:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Channel power and ACPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:DELTA:Y?
```

## Related Commands

CALCulate:MARKer:ADD, CALCulate:ACPower:MARKer<x>:DELTA:X?

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCULATE:ACPOWER:MARKER1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## CALCulate:ACPower:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Channel power and ACPR measurement

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

CALCULATE:ACPOWER:MARKER1:MAXIMUM moves Marker 1 (M1) to the highest peak on the trace.

## CALCulate:ACPower:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:ACPower:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCULATE:ACPOWER:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:ACPower:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:ACPower:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCULATE:ACPOWER:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:ACPower:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:X <value>
```

```
CALCulate:ACPower:MARKer<x>:X?
```

## Related Commands

```
CALCulate:ACPower:MARKer<x>:Y?
```

## Arguments

<value> ::= <NRf> specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

```
CALCULATE:ACPOWER:MARKER1:X 800MHz places Marker 1 (M1) at 800 MHz on the trace.
```

## CALCulate:ACPower:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Calculate commands

## Syntax

```
CALCulate:ACPower:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:ACPower:MARKer<x>:X
```

## Arguments

None

## Returns

<NRf>Marker amplitude of the selected marker.

## Examples

```
CALCULATE:ACPOWER:MARKER1:Y might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.
```

## CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker time for the selected marker in the AM/FM/PM measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Calculate commands

### Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:X?
```

### Related Commands

```
CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:Y?
```

### Arguments

None

### Returns

<NRf> Delta marker time for the selected marker.

### Examples

```
CALCULATE:AM:MARKER1:DELTA:X? might return 38.0E-6, indicating that the delta marker time is 38.0 μs.
```

## CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the AM/FM/PM measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Calculate commands

### Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:Y?
```

## Related Commands

CALCulate:{AM|FM|PM}:MARKer<x>:DELTA:X?

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker, indicating the difference of modulation factor in percent (AM), frequency deviation in Hz (FM), or phase deviation in degrees (PM) with the reference marker.

## Examples

CALCULATE:AM:MARKER1:DELTA:Y? might return 45.82, indicating that the delta marker amplitude is the modulation factor difference of 45.82% in the AM measurement.

## CALCulate:{AM|FM|PM}:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

CALCulate:{AM|FM|PM}:MARKer<x>:MAXimum

## Arguments

None

## Examples

CALCULATE:AM:MARKER1:MAXIMUM moves Marker 1 (M1) to the highest peak on the AM-demodulated trace.

## CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the trace in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands



## Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:HIGHer
```

## Related Commands

```
CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALCULATE:AM:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the AM-demodulated trace.

## CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCULATE:AM:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the AM-demodulated trace.

## CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the trace in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

`CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LOWer`

## Related Commands

`CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:HIGHer`

## Arguments

None

## Examples

`CALCULATE:AM:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the AM-demodulated trace.

## **CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

`CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:RIGHT`

## Related Commands

`CALCulate:{AM|FM|PM}:MARKer<x>:PEAK:LEFT`

## Arguments

None

## Examples

`CALCULATE:AM:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the AM-demodulated trace.

## **CALCulate:{AM|FM|PM}:MARKer<x>:X**

Sets or queries the horizontal position (time) of the selected marker in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:X <value>
```

```
CALCulate:{AM|FM|PM}:MARKer<x>:X?
```

## Related Commands

```
CALCulate:{AM|FM|PM}:MARKer<x>:Y?
```

## Arguments

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

## Examples

```
CALCULATE:AM:MARKER1:X 1.5us places Marker 1 (M1) at 1.5 μs on the AM-demodulated trace.
```

## CALCulate:{AM|FM|PM}:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the AM/FM/PM measurement.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Calculate commands

## Syntax

```
CALCulate:{AM|FM|PM}:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:{AM|FM|PM}:MARKer<x>:X
```

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker, indicating the modulation factor in percent (AM), frequency deviation in Hz (FM), or phase deviation in degrees (PM) at the marker.

## Examples

`CALCULATE:AM:MARKER1:Y?` might return 23.4, indicating that Marker 1 (M1) reads the modulation factor of 23.4% in the AM measurement.

## CALCulate:AUDio:HARMonic:HNUMber

Sets or queries the number of harmonics used for the audio measurement. The range of values is an integer from 1 to 20.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:AUDio:HARMonic:HNUMber <value>`

`CALCulate:AUDio:HARMonic:HNUMber?`

### Arguments

`<value> ::= <NR1>` specifies the number of harmonics.

## Examples

`CALCULATE:AUDIO:HARMONIC:HNUMBER 20` sets the number of harmonics to 20.

## CALCulate:AUDio:HARMonic:NHNumber

Sets or queries the number of non-harmonics used for the audio measurement. The range of values is an integer from 0 to 20.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:AUDio:HARMonic:NHNumber <value>`

`CALCulate:AUDio:HARMonic:NHNumber?`

### Arguments

`<value> ::= <NR1>` specifies the number of non-harmonics.

## Examples

`CALCULATE:AUDIO:HARMONIC:NHNUMBER 20` sets the number of non-harmonics to 20.

## CALCulate:AUDio:NHARmonic:EXCursion

Sets or queries the non-harmonic excursion value. The range of values is 0 dB to 30 dB.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

```
CALCulate:AUDio:NHARmonic:EXCursion <value>
```

```
CALCulate:AUDio:NHARmonic:EXCursion?
```

### Arguments

`<value> ::= <NR1>`The specifies the non-harmonic excursion value.

## Examples

`CALCULATE:AUDIO:NHARMONIC:EXCURSION 6` sets the non-harmonic excursion to 6.0 dB.

## CALCulate:AUDio:NHARmonic:IGNore

Sets or queries the ignore region of the non-harmonics in the audio measurement. The range of the ignore region is from 0 Hz to the audio bandwidth.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

```
CALCulate:AUDio:NHARmonic:IGNore <value>
```

```
CALCulate:AUDio:NHARmonic:IGNore?
```

### Arguments

`<value> ::= <NRf>`specifies the ignore region.

## Examples

`CALCULATE:AUDIO:NHARMONIC:IGNORE 1E+3` sets the non-harmonic ignore region to 1.0 kHz.

## CALCulate:AUDio:NHARmonic:THReshold

Sets or queries the non-harmonic threshold value for the audio measurement. The range is from -200 dBc to 0 dBc.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:AUDio:NHARmonic:THReshold <value>`

`CALCulate:AUDio:NHARmonic:THReshold?`

### Arguments

`<value> ::= <NR1>` specifies the threshold value.

## Examples

`CALCULATE:AUDIO:NHARMONIC:THRESHOLD -100` sets the non-harmonic threshold value to -100 dBc.

## CALCulate:AUDio:SPECtrum:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the selected marker on the spectrum trace.

The parameter `<x>` = 1 to 4. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Conditions

Measurement views: Audio Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:AUDio:SPECtrum:MARKer<x>:DELTA:X?`

### Arguments

None

### Returns

`<NRf>` the delta marker frequency for the selected marker.

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1:DELta:X?` might return `5.0020E+3` indicating the delta marker frequency is 5.002 kHz.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:DELta:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the spectrum trace.

The parameter `<x>` = 1 to 4. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:DELta:Y?`

## Related Commands

`CALCulate:AUDio:SPECTrum:MARKer<x>:DELta:X?`

## Arguments

None

## Returns

`<NRf>` the delta marker amplitude for the selected marker.

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1:DELta:Y?` might return `3.435` indicating that the delta marker amplitude is 3.435 dB.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:HIGHer`

## Related Commands

`CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next higher peak in amplitude on the trace.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LEFT`



---

## Related Commands

CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Examples

CALCulate:AUDio:SPECTrum:MARKer1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LOWer

## Related Commands

CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:HIGHer

## Arguments

None

## Examples

CALCulate:AUDio:SPECTrum:MARKer1:PEAK:LOWer moves Marker 1 (M1) to the next lower peak in amplitude on the trace.

## CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:RIGHT`

## Related Commands

`CALCulate:AUDio:SPECTrum:MARKer<x>:PEAK:LEFT`

## Arguments

None

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:TRACe**

Sets or queries the trace on which the specified marker is placed in the spectrum measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:AUDio:SPECTrum:MARKer<x>:TRACe TRACE1`

`CALCulate:AUDio:SPECTrum:MARKer<x>:TRACe?`

## Arguments

`TRACE1` places the specified marker on Trace 1.

## Examples

`CALCulate:AUDio:SPECTrum:MARKer1 TRACE1` places Marker 1 (M1) on Trace 1.

## **CALCulate:AUDio:SPECTrum:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:AUDio:SPECTrum:MARKer<x>:X <value>
```

```
CALCulate:AUDio:SPECTrum:MARKer<x>:X?
```

## Related Commands

```
CALCulate:AUDio:SPECTrum:MARKer<x>:Y?
```

## Arguments

<value> ::= <NRf> specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

```
CALCulate:AUDio:SPECTrum:MARKer1:X 800 places Marker 1 (M1) at 800 Hz on the spectrum trace.
```

## CALCulate:AUDio:SPECTrum:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker on the spectrum trace.

## Conditions

Measurement views: Audio Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:AUDio:SPECTrum:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:AUDio:SPECTrum:MARKer<x>:X
```

## Arguments

None

## Returns

<NRf> the marker amplitude for the selected marker.

## Examples

```
CALCulate:AUDio:SPECTrum:MARKer1:Y? might return 53.818 indicating that Marker 1 (M1) is at 53.818 dBm.
```

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker time for the selected marker in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Related Commands

CALCulate[:MEASview<y>]:MARKer<x>:ADD, CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:DELTA:X? might return 38.0E-9, indicating that the delta marker time is 38.0 ns on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

CALCulate[:MEASview<y>]:MARKer:ADD, CALCulate:AVTime[:MEASview<y>]:MARKer<x>:DELTA:X?

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time with specified Measurement view number.

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:MAXIMUM moves Marker 1 (M1) to the highest peak on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the Amplitude versus Time trace.

## Conditions

Measurement views: Amplitude versus Time with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
CALCULATE:AVTIME:MEASVIEW1:MARKER1:PEAK:HIGHER
```

 moves Marker 1 (M1) to the next peak higher in amplitude on the trace of Amplitude versus Time Measurement View 1.

## **CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`CALCulate:AVTime:MEASVIEW1:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace of Amplitude versus Time Measurement View 1.

## **CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the Amplitude versus Time trace

### Conditions

Measurement views: Amplitude versus Time with specified Measurement view number.

### Group

Calculate commands

### Syntax

`CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

### Related Commands

`CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

### Arguments

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command

The parameter `<y>`=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Examples

`CALCULATE:AVTIME:MEASVIEW1:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace of Amplitude versus Time Measurement View 1.

## **CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the Amplitude versus Time measurement.

### Conditions

Measurement views: Amplitude versus Time

### Group

Calculate commands

### Syntax

`CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT`

## Related Commands

CALCulate:AVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:TRACe

Sets or queries the trace on which the specified marker is placed in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3 | TRACE4 }
```

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:TRACe?
```

## Arguments

TRACE1 places the selected marker on Trace 1.

TRACE2 places the selected marker on Trace 2.

TRACE3 places the selected marker on Trace 3.

TRACE4 places the selected marker on Trace 4 (math trace).

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:TRACE1 places Marker 1 (M1) on Trace 1 of Amplitude versus Time Measurement View 1.



## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Amplitude versus Time measurement.

### Conditions

Measurement views: Amplitude versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:X?
```

### Related Commands

CALCulate:AVTime[:MEASview<y>]:MARKer<x>:Y?

### Arguments

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:X 1.5U places Marker 1 (M1) at 1.5  $\mu$ s on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:AVTime[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Amplitude versus Time measurement.

### Conditions

Measurement views: Amplitude versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:AVTime[:MEASview<y>]:MARKer<x>:Y?
```

### Related Commands

CALCulate:AVTime[:MEASview<y>]:MARKer<x>:X

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

CALCULATE:AVTIME:MEASVIEW1:MARKER1:Y? might return -2.73, indicating Marker 1 (M1) is at -2.73 dBm on the trace of Amplitude versus Time Measurement View 1.

## CALCulate:BLUEtooth:CONSte:MARKer<x>:DELTA:X[:TIME]? (Query only)

Queries the delta marker time for the selected marker on the Bluetooth display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

CALCulate:BLUEtooth:CONSte:MARKer<x>:DELTA:X[:TIME]?

## Arguments

None

## Returns

<NRf> Delta marker time for the selected marker.

Use [SENSe]:BLUEtooth:ANALysis:TIME:UNITs to select the time unit: symbols or seconds.

## Examples

CALCulate:BLUEtooth:CONSte:MARKer1:DELTA:X:TIME? might return 62.75, indicating that the delta marker time is 62.75 symbols.

## CALCulate:BLUEtooth:CONSte:MARKer<x>:FDEVIation? (Query Only)

Queries the frequency deviation for the selected marker in the Bluetooth Constellation measurement.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth Constellation.

Applicable when the Trace Type is set to Freq Dev.

## Group

Calculate commands

## Syntax

```
CALCulate:BLUETOOTH:CONStE:MARKer<x>:FDEVIation?
```

## Related Commands

The following commands are applicable when the Trace Type is set to IQ:

```
CALCulate:BLUETOOTH:CONStE:MARKer<x>:MAGNitude?
```

```
CALCulate:BLUETOOTH:CONStE:MARKer<x>:PHASe?
```

Use " " to change the Trace Type.

## Arguments

None

## Returns

<NRf> The frequency deviation readout for the selected marker.

## Examples

CALCulate:BLUETOOTH:CONStE:MARKer1:FDEVIation? Might return 102.824E+3, indicating the frequency deviation readout of Marker 1 (M1) is 102.824 kHz.

## CALCulate:BLUETOOTH:CONStE:MARKer<x>:MAGNitude? (Query Only)

Queries the magnitude readout of the selected marker in the Bluetooth Constellation.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth Constellation

Applicable when the Trace Type is set to IQ.

## Group

Calculate commands

## Syntax

`CALCulate:BLUETOOTH:CONStE:MARKer<x>:MAGNitude?`

## Arguments

None

## Returns

<NRf> The magnitude for the selected marker.

## Examples

`CALCulate:BLUETOOTH:CONStE:MARKer1:MAGNitude?` might return `923E-3`, indicating the magnitude readout of Marker 1 (M1) is `923E-3`.

## **CALCulate:BLUETOOTH:CONStE:MARKer<x>:MAXimum (No Query Form)**

Positions the selected marker at the symbol in the center of the analyzed time record.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

`CALCulate:BLUETOOTH:CONStE:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCulate:BLUETOOTH:CONStE:MARKer1:MAXimum` positions Marker 1 (M1) at the symbol in the center of the time record.

## **CALCulate:BLUETOOTH:CONStE:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker in the time domain to the next symbol number to the left, relative to the previous marker position.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:CONSte:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:BLUEtooth:CONSte:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCulate:BLUEtooth:CONSte:MARKer1:PEAK:LEFT` moves Marker 1 (M1) in the time domain to the next lower symbol number.

## CALCulate:BLUEtooth:CONSte:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker in the time domain to the next higher symbol number, relative to the previous marker position.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:CONSte:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:BLUEtooth:CONSte:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCulate:BLUEtooth:CONSte:MARKer1:PEAK:RIGHT` moves the Marker 1 (M1) in the time domain to the next higher symbol number.

## **CALCulate:BLUetooth:CONSte:MARKer<x>:PHASe? (Query Only)**

Queries the phase readout of the selected marker in the Bluetooth Constellation.

### **Conditions**

Measurement views: Bluetooth Constellation.

Applicable when the Trace Type is set to IQ.

### **Group**

Calculate commands

### **Syntax**

`CALCulate:BLUetooth:CONSte:MARKer<x>:PHASe?`

### **Arguments**

None

### **Returns**

<NRf>

The phase readout for the selected marker in degrees between -180 and +180.

### **Examples**

`CALCulate:BLUetooth:CONSte:MARKer1:PHASe?` might return 35.74, indicating the phase readout of Marker 1 (M1) is 35.74°.

## **CALCulate:BLUEetooth:CONSte:MARKer<x>:SYMBol? (Query Only)**

Queries the symbol readout value for the selected marker in the Bluetooth Constellation measurement.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### **Conditions**

Measurement views: Bluetooth Constellation

### **Group**

Calculate commands

### **Syntax**

`CALCulate:BLUEetooth:CONSte:MARKer<x>:SYMBol?`

### **Related Commands**

`CALCulate:BLUEetooth:CONSte:MARKer<x>:X`

## Arguments

None

## Returns

<NRf> The symbol readout for the selected marker.

## Examples

`CALCulate:BLUEtooth:CONSte:MARKer1:SYMBOL?` might return 62.00, indicating the Symbol readout of Marker 1 (M1) is 62.

## CALCulate:BLUEtooth:CONSte:MARKer<x>:Trace

Sets or queries the Trace for the selected marker on the Bluetooth Constellation display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:CONSte:MARKer<x>:Trace <Enum>`

`CALCulate:BLUEtooth:CONSte:MARKer<x>:Trace?`

## Arguments

<Enum> ::= {TRACE1 | TRACE2}. The arguments specify which trace to place the marker on.

## Examples

`CALCulate:BLUEtooth:CONSte:MARKer1:TRACe TRACE1` places Marker 1 (M1) on the Trace 1.

## CALCulate:BLUETooth:CONSte:MARKer<x>:VALue? (Query Only)

Queries the value readout of the selected marker in the Bluetooth Constellation.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

`CALCulate:BLUETOOTH:CONStE:MARKer<x>:VALue?`

## Arguments

None

## Returns

<NRf> The value readout for the selected marker.

## Examples

`CALCulate:BLUETOOTH:CONStE:MARKer2:VALue?` might return 1.00, indicating the value readout of Marker 2(M2) is 1.

## **CALCulate:BLUETOOTH:CONStE:MARKer<x>:X**

Sets or queries the time position of the selected marker on the Bluetooth Constellation.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Calculate commands

## Syntax

`CALCulate:BLUETOOTH:CONStE:MARKer<x>:X <value>`

`CALCulate:BLUETOOTH:CONStE:MARKer<x>:X?`

## Arguments

<value> ::= <NRf> specifies the time position of the marker. Range (analysis offset) to [(analysis offset) + (analysis length)].

## Examples

`CALCulate:BLUETOOTH:CONStE:MARKer1:X -1.63875ms` places the Marker 1 (M1) at -1.63875 ms on the CONSTellation trace.

## **CALCulate:BLUETOOTH:EDIagram:MARKer<x>:DELTA:X[:TIME]? (Query Only)**

Returns the delta marker time for the selected marker on the Bluetooth eye diagram display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.



## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:EDIagram:MARKer<x>:DELTA:X[:TIME]?
```

## Related Commands

```
CALCulate:BLUEtooth:EDIagram:MARKer<x>:DELTA:Y?
```

## Arguments

None

## Returns

<NRf> the delta marker time for the selected marker.

Use [SENSe]:BLUEtooth:ANALysis:TIME:UNITs to select the time unit: symbols or seconds.

## Examples

CALCulate:BLUEtooth:EDIagram:MARKer1:DELTA:X:TIME? might return 62.75, indicating that the delta marker time is 62.75 symbols.

## CALCulate:BLUEtooth:EDIagram:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected markers in the Bluetooth eye diagram.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:EDIagram:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:BLUEtooth:EDIagram:MARKer<x>:DELTA:X[:TIME]?
```

## Returns

<NRf>

the delta marker amplitude for the selected marker. Units are in Hz if the Trace Type is set to Freq Dev and no units if the Trace Type is set to IQ.

## Examples

`CALCulate:BLUEtooth:EDIagram:MARKer1:DELta:Y?` might return -1.043, indicating that the delta marker amplitude is -1.043.

## **CALCulate:BLUEtooth:EDIagram:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the Bluetooth eye diagram trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCulate:BLUEtooth:EDIagram:MARKer1:MAXimum` moves Marker 1 (M1) to the Highest peak on the trace.

## **CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Bluetooth eye diagram trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:HIGHer`

## Related Commands

CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LOWer

## Arguments

None

## Examples

CALCulate:BLUEtooth:EDIagram:MARKer1:PEAK:HIGHer moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the Bluetooth eye diagram trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LEFT
```

## Related Commands

CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Examples

CALCulate:BLUEtooth:EDIagram:MARKer1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Bluetooth eye diagram trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LOWer`

## Related Commands

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:HIGHer`

## Arguments

None

## Examples

`CALCulate:BLUEtooth:EDIagram:MARKer1:PEAK:LOWer` moves Marker 1(M1) to the next peak lower in amplitude on the trace.

## **CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the Bluetooth eye diagram trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:RIGHT`

## Related Commands

`CALCulate:BLUEtooth:EDIagram:MARKer<x>:PEAK:LEFT`

## Arguments

None

## Examples

`CALCulate:BLUEtooth:EDIagram:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:BLUetooth:EDIagram:MARKer<x>:TRACe

The command places the selected marker on the I or Q trace in the Bluetooth eye diagram display. The query returns the name of the trace where the current marker resides.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADDcommand.

### Conditions

Measurement views: Bluetooth eye diagram

### Group

Calculate commands

### Syntax

```
CALCulate:BLUetooth:EDIagram:MARKer<x>:TRACe <Enum>
```

```
CALCulate:BLUetooth:EDIagram:MARKer<x>:TRACe?
```

### Arguments

```
<Enum> ::= {TRACE1 | TRACE2}.
```

The arguments specify which trace to place the marker on.

Trace1 specifies the I trace and Trace2 specifies the Q trace.

### Returns

For I and Q traces, TRACE1 identifies the I trace and TRACE2 identifies the Q trace.

### Examples

```
CALCulate:BLUetooth:EDIagram:MARKer1:TRACe TRACE1 places Marker 1 (M1) on the I trace.
```

## CALCulate:BLUetooth:EDIagram:MARKer<x>:X[:TIME]

Sets or queries the horizontal position (time) of the selected marker in the Bluetooth eye diagram measurement.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: Bluetooth eye diagram

### Group

Calculate commands

### Syntax

```
CALCulate:BLUetooth:EDIagram:MARKer<x>:X[:TIME] <value>
```

```
CALCulate:BLUetooth:EDIagram:MARKer<x>:X[:TIME]?
```

## Related Commands

CALCulate:BLUEtooth:EDIagram:MARKer<x>:Y?

## Arguments

<value> ::= <NRf> the horizontal position (time) of the marker.

Use [SENSe]:BLUEtooth:ANALysis:TIME:UNITs to select the time unit: symbols or seconds.

## Examples

CALCulate:BLUEtooth:EDIagram:MARKer1:X:TIME 38.5 places Marker 1 (M1) at 38.5 symbols on the trace.

## CALCulate:BLUEtooth:EDIagram:MARKer<x>:Y? (Query Only)

Returns the vertical position of the selected marker in the Bluetooth Eye diagram measurement.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Bluetooth eye diagram

## Group

Calculate commands

## Syntax

CALCulate:BLUEtooth:EDIagram:MARKer<x>:Y?

## Related Commands

CALCulate:BLUEtooth:EDIagram:MARKer<x>:X[:TIME]

## Arguments

None

## Returns

<value> ::= <NRf> the vertical position of the selected marker. The units are Hz if the Trace Type is set to Freq Dev; if the Trace Type is set to IQ, there are no units.

## Examples

CALCulate:BLUEtooth:EDIagram:MARKer1:Y? Might return 571.8E-3, indicating Marker 1 (M1) is at 0.5718.

## CALCulate:BLUEtooth:FDVTIME:MARKer<x>:DELTA:X[:TIME]? (Query Only)

Queries the delta marker time for the selected marker on the Frequency Deviation vs. Time display

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Use [SENSe]:BLUETOOTH:ANALYSIS:TIME:UNITS to select the time unit symbols or seconds.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUETOOTH:FDVTIME:MARKer<x>:DELTA:X[:TIME]?
```

## Related Commands

```
CALCulate:BLUETOOTH:FDVTIME:MARKer<x>:DELTA:Y?
```

## Arguments

None

## Returns

<NRf> the delta marker time for the selected marker.

## Examples

CALCULATE:BLUETOOTH:FDVTIME:MARKER1:DELTA:X:TIME? might return 62.75, indicating that the delta marker time is 62.75 symbols.

## CALCulate:BLUETOOTH:FDVTIME:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUETOOTH:FDVTIME:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:BLUETOOTH:FDVTIME:MARKer<x>:DELTA:X[:TIME]?
```

## Arguments

None

## Returns

<NRf> the delta marker amplitude for the selected marker.

## Examples

`CALCulate:BLUEtooth:FDVTime:MARKer1:DELta:Y?` return -1.043, indicating that the delta marker amplitude is -1.043.

## **CALCulate:BLUEtooth:FDVTime:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the trace in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:FDVTime:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCULATE:BLUETOOTH:FDVTIME:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:HIGHer`



---

## Related Commands

CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LOWer

## Arguments

None

## Examples

CALCULATE:BLUETOOTH:FDVTIME:MARKER1:PEAK:HIGHER moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LEFT

## Related Commands

CALCulate:BLUEtooth:CONSt:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Examples

CALCULATE:BLUETOOTH:FDVTIME:MARKER1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:HIGHer
```

## Arguments

None

## Examples

`CALCULATE:BLUETOOTH:FDVTIME:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:BLUEtooth:FDVTime:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCULATE:BLUETOOTH:FDVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:BLUEtooth:FDVTIME:MARKer<x>:X[:TIME]**

Sets or queries the marker time in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4. The specified marker must be activated using the CALCulate:MARKer:ADD command.

Use [SENSe]:BLUEtooth:ANALysis:TIME:UNITs to select the time unit symbols (default) or seconds.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:X[:TIME] <value>
```

```
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:X[:TIME]?
```

## Related Commands

CALCulate:BLUEtooth:FDVTIME:MARKer<x>:Y

## Arguments

<value> ::= <NRf> specifies the marker time in seconds or in symbols based on the Units specified under the Analysis Time.

## Examples

```
CALCULATE:BLUETOOTH:FDVTIME:MARKER1:X:TIME -234.5E-6 places Marker 1 (M1) at -234.5 µs on the trace.
```

## CALCulate:BLUEtooth:FDVTIME:MARKer<x>:Y

Sets or queries the marker amplitude in the Frequency Deviation vs. Time display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:BLUEtooth:FDVTIME:MARKer<x>:Y
```

## Related Commands

CALCulate:BLUEtooth:FDVTIME:MARKer<x>:X[:TIME]

## Arguments

<value> ::= <NRf> specifies the marker frequency value in Hz.

## Examples

`CALCULATE:BLUETOOTH:FDVTIME:MARKER1:Y?` might return `571.8E-3`, indicating Marker 1 (M1) is 0.5718 Hz.

## **CALCulate:BOBW:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker frequency for the selected marker on the Bluetooth 20 dB bandwidth measurement.

The parameter `<x>` = 1 to 4; Marker 0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

`CALCulate:BOBW:MARKer<x>:DELTA:X?`

## Related Commands

`CALCulate:BOBW:MARKer<x>:DELTA:Y?`

## Arguments

None

## Returns

`<NRf>`, Delta marker frequency for the selected marker in Hz.

## Examples

`CALCULATE:BOBW:MARKER1:DELTA:X?` might return `62.75E+3`, indicating that the delta marker time is 62.75 kHz.

## **CALCulate:BOBW:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the Bluetooth 20 dB bandwidth measurement.

The parameter `<x>` = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:BOBW:MARKer<x>:DELTA:X?
```

## Arguments

None

## Returns

<NRf>, the Delta marker amplitude for the selected marker.

## Examples

`CALCULATE:BOBW:MARKER1:DELTA:Y?` might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## CALCulate:BOBW:MARKer<x>:MAXimum (No Query Form)

Positions the selected marker at the maximum value of the display.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCULATE:BOBW:MARKER1:MAXIMUM` positions Marker 1 (M1) at the maximum value of the display.

## CALCulate:BOBW:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the BT 20dB bandwidth trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:PEAK:HIGHer
```

## Related Commands

```
CALCulate:BOBW:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALCULATE:BOBW:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:BOBW:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker in the time domain to the left, relative to the previous marker position on the trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCULATE:BOBW:MARKER1:PEAK:LEFT` moves the Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:BOBW:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the BT 20dB bandwidth trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

CALCULATE:BOBW:MARKER1:PEAK:LOWER moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:BOBW:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker in the time domain to the right, relative to the previous marker position on the trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

```
CALCulate:BOBW:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

CALCULATE:BOBW:MARKER1:PEAK:RIGHT moves the Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:BOBW:MARKer<x>[:SET]:CENTER (No Query Form)

Sets the center frequency to the value at the marker position in the BT 20 dB Bandwidth view.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: BT 20 dB Bandwidth

### Group

Calculate commands

### Syntax

```
CALCulate:BOBW:MARKer<x>[:SET]:CENTer
```

### Arguments

None

### Examples

CALCULATE:BOBW:MARKER1:SET:CENTER sets the center frequency to the value at the Marker 1 (M1) position.

## CALCulate:BOBW:MARKer<x>:X

Sets or queries the time position of the selected marker on the BT 20 dB Bandwidth measurement.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: BT 20 dB Bandwidth

### Group

Calculate commands

### Syntax

```
CALCulate:BOBW:MARKer<x>:X <value>
```

```
CALCulate:BOBW:MARKer<x>:X?
```

### Related Commands

```
CALCulate:BOBW:MARKer<x>:Y?
```

### Arguments

<value> ::= <NRf> specifies the horizontal position of the marker.

Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").



## Examples

`CALCULATE:BOBW:MARKER1:X 2.480000000E+9` places the Marker 1 (M1) at 2.480000000E+9 on the BT 20dB Bandwidth trace.

## CALCulate:BOBW:MARKer<x>:Y? (Query Only)

Returns the amplitude position of the selected marker on the Bluetooth 20 dB Bandwidth trace.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: BT 20 dB Bandwidth

## Group

Calculate commands

## Syntax

`CALCulate:BOBW:MARKer<x>:Y?`

## Related Commands

`CALCulate:BOBW:MARKer<x>:DELta:X?`

## Arguments

None

## Returns

<value> ::= <NRf> specifies the vertical position of the marker.

## Examples

`CALCULATE:BOBW:MARKER1:Y?` returns the marker amplitude in dB on the BT 20dB Bandwidth trace.

## CALCulate:CONSte[:MEASview<y>]:MARKer<x>:DELta:X[:TIME]? (Query only)

Returns the delta marker time for the selected marker on the Constellation trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?`

## Returns

<NRf> Delta marker time for the selected marker.

The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:DELTA:X:TIME?` might return `-62.75`, indicating that the delta marker time is `-62.75` symbols for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:FDEVIation? (Query only)**

Queries the frequency deviation of the selected marker in the Constellation measurement for an FSK modulated signal.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

The command is valid for 2, 4, 8, 16 FSK or C4FM modulated signals.

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:FDEVIation?`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:MAGNitude?](#)

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:PHASe?](#)

## Arguments

None

## Returns

<NRf> the frequency deviation readout for the selected marker.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:FDEVIation?` might return `102.824E+3`, indicating the frequency deviation readout of Marker 1 (M1) is 102.824 kHz for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAGNitude? (Query only)**

Queries the magnitude readout of the selected marker in the Constellation measurement.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

`MEASview` keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: `GPDemod Constellation`).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAGNitude?`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

None

## Returns

`<NRf>` the magnitude readout for the selected marker.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:MAGNitude?` might return `0.713927`, indicating the magnitude readout of Marker 1 (M1) is 0.713927 for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Positions the selected marker at the symbol in the center of the time record.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created/number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

```
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:MAXimum` positions Marker 1 (M1) at the symbol in the center of the time record for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker in the time domain to the next lower symbol number, relative to the previous marker position.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

```
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[\*CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT\*](#)

## Arguments

None

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:PEAK:LEFT` moves Marker 1 (M1) in the time domain to the next lower symbol number for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker in the time domain to the next higher symbol number, relative to the previous marker position.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PEAK:RIGHT`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:PEAK:RIGHT` moves the Marker 1 (M1) in the time domain to the next higher symbol number for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PHASe? (Query only)**

Queries the phase readout of the selected marker in the Constellation measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:PHASe?`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

None

## Returns

<NRf> The phase readout for the selected marker.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:PHASe` might return 35.74, indicating the phase readout of Marker 1 (M1) is 35.74 ° for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:SYMBOL? (Query only)**

Queries the symbol readout of the selected marker in the Constellation measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:SYMBOL?`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

None

## Returns

<NRf> the symbol readout for the selected marker.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:SYMBOL?` might return 62.00, indicating the symbol readout of Marker 1 (M1) is 62 for measview 2 of constellation.

## **CALCulate:CONSte[:MEASview<y>]:MARKer<x>:VALue? (Query only)**

Queries the value readout of the selected marker in the Constellation measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Calculate commands

## Syntax

`CALCulate:CONSte[:MEASview<y>]:MARKer<x>:VALue?`

## Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

None

## Returns

<NRf> the value readout for the selected marker.

## Examples

`CALCulate:CONSte:MEASview2:MARKer1:VALue?` might return 2.00, indicating the value readout of Marker 1 (M1) is 2 for measview 2 of constellation.

## CALCulate:CONSte[:MEASview<y>]:MARKer<x>:X

Sets or queries the time position of the selected marker on the Constellation trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

### Group

Calculate commands

### Syntax

```
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:X<value>
```

```
CALCulate:CONSte[:MEASview<y>]:MARKer<x>:X?
```

### Related Commands

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:MAGNitude?](#)

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:PHASe?](#)

[CALCulate:CONSte\[:MEASview<y>\]:MARKer<x>:SYMBol?](#)

### Arguments

<value>::=<NRf> specifies the time position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

### Examples

CALCulate:CONSte:MEASview2:MARKer1:X -1.63875 m places the Marker 1 (M1) at -1.63875 ms on the Constellation trace for measview 2 of constellation.

## CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]? (Query only)

Returns the delta marker time for the selected marker on the Demodulated I&Q versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for MSO58 channels).

### Conditions

Measurement views: Demodulated I&Q versus Time



MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?
```

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

<NRf> Delta marker time for the selected marker.

Use the [SENSe]:DDEMod:TIME:UNITs command to select the time unit, symbols (default) or seconds.

## Examples

CALCulate:DIQVtime:MEASview1:MARKer1:DELTA:X:TIME? might return 62.75, indicating that the delta marker time is 62.75 symbols measurement view 1.

## CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker amplitude for the selected marker on the Demodulated I&Q versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:DELTA:X\[:TIME\]?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in volts.

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:DELTA:Y?` might return `-78.24E-3`, indicating that the delta marker amplitude is `-78.24` mV measurement view 1.

## **CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the Demodulated I&Q versus Time trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace measurement view 1.

## **CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the Demodulated I&Q versus Time trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer](#)

## Arguments

None

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace measurement view 1.

## **CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the Demodulated I&Q versus Time trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace measurement view 1.

## **CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the Demodulated I&Q versus Time trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

[\*CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer\*](#)

## Arguments

None

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace measurement view 1.

## **CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the Demodulated I&Q versus Time trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace measurement view 1.

## CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:TRACe

Places the selected marker on the Demodulated I&Q versus Time trace. The query returns the name of the trace on which the marker resides.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:TRACe <Enum>
```

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:TRACe?
```

## Related Commands

[CALCulate:IQVTime:MARKer<x>:TRACe](#)

## Arguments

`<Enum> ::= TRACE1 | TRACE2` specifies which trace to place the marker on. TRACE1 specifies the “I” trace and TRACE2 specifies the “Q” trace.

## Returns

For I and Q traces, TRACE1 identifies the “I” trace and TRACE2 identifies the “Q” trace.

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:TRACe` TRACE1 places Marker 1 (M1) on the I trace measurement view 1.

## CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:X[:TIME]

Sets or queries the horizontal position (time) of the selected marker in the Demodulated I&Q versus Time measurement.

The parameter `<x>` = Marker 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:X[:TIME] <value>`

`CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:X[:TIME]?`

## Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:Y?](#)

## Arguments

`<value> ::= <NRf>` specifies the horizontal position (time) of the marker. Use the `[SENSe]:DDEMod:TIME:UNITs` command to select the time unit, symbols (default) or seconds.

## Examples

`CALCulate:DIQVtime:MEASview1:MARKer1:X:TIME 38.5` places Marker 1 (M1) at 38.5 symbols on the trace measurement view 1.

## CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:Y? (Query only)

Queries the marker amplitude of the selected marker in the Demod I&Q versus Time measurement.

The parameter <x> = Marker 0 to 4. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:DIQVtime[:MEASview<y>]:MARKer<x>:Y?
```

### Related Commands

[CALCulate:DIQVtime\[:MEASview<y>\]:MARKer<x>:X\[:TIME\]](#)

### Arguments

None

### Returns

<NRf> Marker amplitude of the selected marker in volts.

### Examples

CALCulate:DIQVtime:MEASview1:MARKer1:Y? might return 25.803E-3, indicating Marker 1 (M1) is at 25.803 mV measurement view 1.

## CALCulate:DPX:MARKer<x>:DELTA:X:AMPLitude? (Query Only)

Returns the delta marker amplitude for the selected marker on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command and attached to the Bitmap trace using the CALCulate:DPX:MARKer<x>:TRACe command.

### Conditions

Measurement views: DPX spectrum

### Group

Calculate commands

## Syntax

CALCulate:DPX:MARKer<x>:DELTA:X:AMPLitude?

## Related Commands

CALCulate:DPX:MARKer<x>:X[:FREQuency], CALCulate:DPX:MARKer<x>:Y?

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in dB.

## Examples

CALCULATE:DPX:MARKER1:DELTA:X:AMPLITUDE? might return 35.12, indicating that the delta marker amplitude is 35.12 dB for Marker 1 (M1).

## CALCulate:DPX:MARKer<x>:DELTA:X[:FREQuency]? (Query Only)

Returns the delta marker frequency for the selected marker on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

CALCulate:DPX:MARKer<x>:DELTA:X[:FREQuency]?

## Related Commands

CALCulate:DPX:MARKer<x>:DELTA:X:AMPLitude?, CALCulate:DPX:MARKer<x>:DELTA:Y?

## Arguments

None

## Returns

<NRf> Delta marker frequency for the selected marker in Hz.



## Examples

`CALCULATE:DPX:MARKER1:DELTA:X?` might return `1.28E+6`, indicating that the delta marker frequency is 1.28 MHz for Marker 1 (M1).

## **CALCulate:DPX:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the DPX spectrum trace.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

`CALCulate:DPX:MARKer<x>:DELTA:Y?`

## Related Commands

`CALCulate:DPX:MARKer<x>:DELTA:X[:FREQuency]?`

## Arguments

None

## Returns

`<NRf>` Delta marker amplitude for the selected marker.

## Examples

`CALCULATE:DPX:MARKER1:DELTA:Y?` might return `23.45`, indicating that the delta marker amplitude is 23.45 dB.

## **CALCulate:DPX:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the DPX spectrum trace.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

`CALCulate:DPX:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCULATE:DPX:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:DPX:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

`CALCulate:DPX:MARKer<x>:PEAK:HIGHer`

## Related Commands

`CALCulate:DPX:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALCULATE:DPX:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:DPX:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:DPX:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCULATE:DPX:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:DPX:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:DPX:MARKer<x>:PEAK:HIGHer
```

## Arguments

None

## Examples

`CALCULATE:DPX:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:DPX:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the DPX spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: DPX spectrum

### Group

Calculate commands

### Syntax

```
CALCulate:DPX:MARKer<x>:PEAK:RIGHT
```

### Related Commands

```
CALCulate:DPX:MARKer<x>:PEAK:LEFT
```

### Arguments

None

### Examples

CALCULATE:DPX:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:DPX:MARKer<x>:POWer:DENSity? (Query Only)

Returns the power density of the specified marker in the DPX measurement.

### Conditions

Measurement view: DPX

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

```
CALCulate:DPX:MARKer<x>:POWer:DENSity?
```

### Returns

<NRf> is the power density of the selected marker in dBm/Hz.

### Examples

CALCULATE:DPX:MARKER1:POWER:DENSITY? might return 80.2432546111, indicating the power density is 80.24 dBm/Hz.

## CALCulate:DPX:MARKer<x>:POWer:IDensity? (Query Only)

Returns the integrated power density reading between the MR (reference marker) and the selected marker in the DPX measurement.

### Conditions

Measurement view: DPX

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

```
CALCulate:DPX:MARKer<x>:POWer:IDensity?
```

### Returns

<NRf> is the integrated power density reading between the MR (reference marker) and the selected marker in dBm.

### Examples

```
CALCULATE:DPX:MARKER1:POWER:REFDENSITY? might return -78.56 dBm/Hz.
```

## CALCulate:DPX:MARKer<x>:POWer:INTEgrated? (Query Only)

Returns the integrated power reading between the MR (reference marker) and the selected marker in the DPX measurement.

### Conditions

Measurement view: DPX

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

```
CALCulate:DPX:MARKer<x>:POWer:INTEgrated?
```

### Returns

<NRf> is the integrated power reading between MR (reference marker) and the selected marker in dBm.

### Examples

```
CALCULATE:DPX:MARKER1:POWER:INTEGRATED? might return -14.66, indicating the integrated power is -14.66 dBm.
```

## **CALCulate:DPX:MARKer<x>[:SET]:CENTer (No Query Form)**

Sets the center frequency to the marker frequency in the DPX spectrum view.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement views: DPX spectrum

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:DPX:MARKer<x>[:SET]:CENTer
```

### **Arguments**

None

### **Examples**

CALCULATE:DPX:MARKER1:SET:CENTER sets the center frequency to the marker frequency in the DPX spectrum view.

## **CALCulate:DPX:MARKer<x>:TRACe**

Sets or queries the trace to attach the specified marker to in the DPX spectrum measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement views: DPX spectrum

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:DPX:MARKer<x>:TRACe { BITMAP | TRACE1 | TRACE2 |TRACE3 | TRACE4 }
```

```
CALCulate:DPX:MARKer<x>:TRACe?
```

### **Arguments**

BITMAP attaches the specified marker to the Bitmap trace.

TRACE1 attaches the specified marker to the +Peak trace.

TRACE2 attaches the specified marker to the -Peak trace.

TRACE3 attaches the specified marker to the Average trace.

TRACE4 attaches the specified marker to the Math trace.

## Examples

CALCULATE:DPX:MARKER1:TRACE TRACE1 attaches Marker 1 (M1) to the +Peak trace.

## CALCulate:DPX:MARKer<x>:X:AMPLitude

Sets or queries the amplitude position of the selected marker in the DPX spectrum view. This command is valid for the marker on the bitmap trace (refer to the CALCulate:DPX:MARKer<x>:TRACe command). The frequency position is set by the CALCulate:DPX:MARKer<x>:X[:FREQuency] command.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

CALCulate:DPX:MARKer<x>:X:AMPLitude <value>

CALCulate:DPX:MARKer<x>:X:AMPLitude?

## Arguments

<value>::=<NRf> specifies the amplitude position of the marker. Range: -100 to 0 dBm.

## Examples

CALCULATE:DPX:MARKER1:X:AMPLITUDE -34.5dBm places Marker 1 (M1) at -34.5 dBm.

## CALCulate:DPX:MARKer<x>:X[:FREQuency]

Sets or queries the frequency position of the selected marker in the DPX spectrum view.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

The command is only valid for the Spectrum and Frequency plots.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:X[:FREQUENCY] <value>
```

```
CALCulate:DPX:MARKer<x>:X[:FREQUENCY] ?
```

## Related Commands

```
CALCulate:DPX:MARKer<x>:Y?
```

## Arguments

<value>::=<NRf> specifies the frequency position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

```
CALCULATE:DPX:MARKER1:X:FREQUENCY 800MHz places Marker 1 (M1) at 800 MHz on the trace.
```

## CALCulate:DPX:MARKer<x>:X:PHASe

Sets or queries the phase position of the selected marker in the DPX phase plot.

The parameter <x> = 0 to 4. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:X:PHASe <value>
```

```
CALCulate:DPX:MARKer<x>:X:PHASe?
```

## Arguments

<value>::=<NRf> phase setting in degrees for the specified marker.

## Examples

```
CALCulate:DPX:MARKer0:X:PHASe? might return 94.0000000 indicating the phase for the reference marker (M0) is 94.00°.
```

## CALCulate:DPX:MARKer<x>:X:TIME

Sets or queries the time position of the selected marker in the DPX spectrum view. The parameter <x> = 0 to 4. The specified marker must be activated using the CALCulate:MARKer:ADD command. The command is valid for the Amplitude, Frequency, and Phase plots.

## Conditions

Measurement views: DPX spectrum



## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:X:TIME <value>
```

```
CALCulate:DPX:MARKer<x>:X:TIME?
```

## Arguments

<value> : : <Nrf> Time position of the specified marker.

## Examples

CALCulate:DPX:MARKer0:X:TIME? might return 1.000000000E-3 indicating the time position of the reference marker (MR0) is 1.000 ms.

## CALCulate:DPX:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker in the DPX spectrum view. The data occurrence rate is returned for the bitmap trace, and the amplitude value for the +peak, -peak, average, and math traces. The horizontal position can be set by the CALCulate:DPX:MARKer<x>:X:AMPLitude and CALCulate:DPX:MARKer<x>:X:FREQuency commands.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: DPX spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:DPX:MARKer<x>:Y?
```

## Arguments

None

## Returns

<Nrf> The value type depends on which trace the marker is placed on (refer to the CALCulate:DPX:MARKer<x>:TRACe command):

The amplitude value is returned in dBm for the marker on the +peak, -peak, average, or math trace. The data occurrence rate is returned in percent (%) for the marker on the bitmap trace.

## Examples

CALCULATE:DPX:MARKer1:Y? might return -34.28 indicating Marker 1 (M1) is at -34.28 dBm when it is placed on the +peak, -peak, average, or math trace.

`CALCULATE:DPX:MARKER1:Y?` might return 76.5 indicating Marker 1 (M1) is at 76.5% when it is placed on the bitmap trace.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]? (Query only)**

Returns the delta marker time for the selected marker on the eye diagram trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of the measurement (Example: Eye Diagram).

### **Group**

Calculate commands

### **Syntax**

`CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?`

### **Related Commands**

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

### **Arguments**

None

### **Returns**

<NRf> Delta marker time for the selected marker. Use the `[SENSE]:DDEMod:TIME:UNITs` command to select the time unit:symbols (default) or seconds.

### **Examples**

`CALCulate:EDIagram:MEASview2:MARKer1:DELTA:X:TIME?` might return 62.75, indicating that the delta marker time is 62.75 symbols for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)**

Returns the delta marker amplitude for the selected marker on the eye diagram trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:DELTA:X\[:TIME\]?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

`CALCulate:EDIagram:MEASview2:MARKer<x>:DELTA:Y?` might returns -1.043, indicating that the delta marker amplitude is -1.043 for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the eye diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye Diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:EDIagram:MEASview1:MARKer<x>:MAXimum` moves the Marker 1 (M1) to the highest peak on the trace for measview1 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the eye diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye Diagram).

## Group

Calculate commands

## Syntax

`CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

## Related Commands

[\*CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer\*](#)

## Arguments

None

## Examples

`CALCulate:EDIagram:MEASview1:MARKer<x>:PEAK:HIGHer` moves the Marker 1 (M1) to the next peak higher in amplitude on the trace for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the eye diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCulate:EDIagram:MEASview2:MARKer<x>:PEAK:LEFT` moves the Marker 1 (M1) to the next peak to the left on the trace for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the eye diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye Diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer](#)

## Arguments

None

## Examples

`CALCulate:EDIagram:MEASview2:MARKer<x>:PEAK:LOWer` moves the Marker 1 (M1) to the next peak lower in amplitude on the trace for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the eye diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:EDIagram:MEASview2:MARKer<x>:PEAK:RIGHT` moves the Marker 1 (M1) to the next peak to the right on the trace for measview2 of the Eye diagram.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:TRACe**

Places the selected marker on the I or Q trace in the eye diagram display. The query returns the name of the trace on which the marker resides. It is valid for all types of modulation except 2|4|8|16 FSK and C4FM.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:TRACe<Enum>
```

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:TRACe?
```

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:X\[:TIME\]](#)

## Arguments

<Enum> ::= TRACE1 | TRACE2 specifies which trace to place the marker on. TRACE1 specifies the “I” trace and TRACE2 specifies the “Q” trace.

## Returns

For I and Q traces, TRACE1 identifies the “I” trace and TRACE2 identifies the “Q” trace.

## Examples

`CALCulate:EDIagram:MEASview2:MARKer1:TRACe TRACE1` places the Marker 1 (M1) on the I trace in an eye diagram for the measview2.

## **CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:X[:TIME]**

Sets or queries the horizontal position (time) of the selected marker in the eye diagram measurement.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye Diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:X[:TIME]<value>
```

`CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:X[:TIME]?`

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:Y?](#)

## Arguments

`<value>::=<NRf>` specifies the horizontal position (time) of the marker. Use the `[SENSe]:DDEMod:TIME:UNITs` command to select the time unit:symbols (default) or seconds.

## Examples

`CALCulate:EDIagram:MEASVIEW2:MARKer1:X:TIME 38.5` places Marker 1 (M1) at 38.5 symbols on the trace for the measview2 of Eye diagram.

## CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:Y? (Query only)

Queries the vertical position of the selected marker in the eye diagram measurement.

The parameter `<x>` = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Calculate commands

## Syntax

`CALCulate:EDIagram[:MEASview<y>]:MARKer<x>:Y?`

## Related Commands

[CALCulate:EDIagram\[:MEASview<y>\]:MARKer<x>:X\[:TIME\]](#)

## Arguments

None

## Returns

`<NRf>` Vertical position of the selected marker.

## Examples

`CALCulate:EDIagram:MEASview2:MARKer<x>:Y?` might return `571.8E-3`, indicating Marker 1 (M1) is at 0.5718 for the measview 2 of Eye diagram.



## CALCulate:EMC:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency value for the selected marker in the EMC view.

<x> = 1,2,3,4 is the marker

### Conditions

Measurement view: EMC - EMI Display

### Group

Calculate commands

### Syntax

```
CALCulate:EMC:MARKer<x>:DELTA:X?
```

### Arguments

None

### Returns

<NRf>

### Examples

CALCulate:EMC:MARKer1:DELTA:X? might return -178.9375, indicating that the delta marker frequency of Marker 1 (M1) is -178.93 KHz.

## CALCulate:EMC:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the EMC view.

<x> = 1,2,3,4 is the marker

### Conditions

Measurement view: EMC - EMI Display

### Group

Calculate commands

### Syntax

```
CALCulate:EMC:MARKer<x>:DELTA:Y?
```

### Arguments

None

### Returns

<NRf>

## Examples

`CALCulate:EMC:MARKer1:DELta:Y?` might return `-2.2589`, indicating that the delta marker Amplitude of Marker 1 (M1) is `-2.25` dB.

## **CALCulate:EMC:MARKer<x>:MAXimum (No Query Form)**

Positions the specified marker at the maximum value of the display on the EMC trace.

`<x>` = 1,2,3,4 is the marker

### Conditions

Measurement view: EMC - EMI Display

### Group

Calculate commands

### Syntax

`CALCulate:EMC:MARKer<x>:MAXimum`

### Arguments

None

### Returns

None

## Examples

`CALCulate:EMC:MARKer1:MAXimum` positions the marker 1 (M1) at the maximum value of the display.

## **CALCulate:EMC:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the specified marker to the next peak higher in amplitude on the EMC trace.

`<x>` = 1,2,3,4 is the marker

### Conditions

Measurement view: EMC - EMI Display

### Group

Calculate commands

### Syntax

`CALCulate:EMC:MARKer<x>:PEAK:HIGHer`

## Arguments

None

## Returns

None

## Examples

`CALCulate:EMC:MARKer2:PEAK:HIGHer` moves Marker 2 (M2) to the next peak higher in amplitude on the trace.

## **CALCulate:EMC:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the specified marker to the next peak to the left on the EMC trace.

<x> = 1,2,3,4 is the marker

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

```
CALCulate:EMC:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Returns

None

## Examples

`CALCulate:EMC:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:EMC:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the specified marker to the next peak lower in amplitude on the EMC trace.

<x> = 1,2,3,4 is the marker

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

`CALCulate:EMC:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Returns

None

## Examples

`CALCulate:EMC:MARKer2:PEAK:LOWer` moves Marker 2 (M2) to the next peak lower in amplitude on the trace.

## **CALCulate:EMC:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the specified marker to the next peak to the right on the EMC trace.

<x> = 1,2,3,4 is the marker

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

`CALCulate:EMC:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Returns

None

## Examples

`CALCulate:EMC:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:EMC:MARKer<x>[:SET]:CENTER (No Query Form)**

Sets the center frequency to the value at the specified marker position in the EMC view.

<x> = 1,2,3,4 is the marker

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

```
CALCulate:EMC:MARKer<x>[:SET]:CENTer
```

## Arguments

None

## Returns

None

## Examples

`CALCulate:EMC:MARKer1[:SET]:CENTer` sets the center frequency to the value at the Marker 1 (M1) position.

## CALCulate:EMC:MARKer<x>:X

Sets or queries the frequency position of the selected marker on the EMC view.

<x> = 1,2,3,4 is the marker

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

```
CALCulate:EMC:MARKer<x>:X <value>
```

```
CALCulate:EMC:MARKer<x>:X ?
```

## Arguments

<value> = <NRf> specifies the horizontal position of the marker.

## Returns

<NRf>

## Examples

`CALCulate:EMC:MARKer1:X` might return `254.562500000E+3`, indicating that the frequency position of the Marker 1 (M1) is 254.56 KHz.

## CALCulate:EMC:MARKer<x>:Y

Sets or queries the amplitude position of the selected marker in the EMC view.

<x> = 1,2,3,4 is the marker

### Conditions

Measurement view: EMC - EMI Display

### Group

Calculate commands

### Syntax

```
CALCulate:EMC:MARKer<x>:Y <value>
```

```
CALCulate:EMC:MARKer<x>:Y?
```

### Arguments

<value> = <NRf> specifies the amplitude position of the marker.

### Returns

<NRf>

### Examples

CALCulate:EMC:MARKer1:Y might return 8.9157, indicating that the amplitude position of Marker 1 (M1) is 8.92 dBuV.

## CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:X? (Query only)

Returns the delta marker time for the selected marker in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58 ).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Related Commands

[CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in percent (%).

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:DELTA:X?` might return 1.62, indicating that the delta marker duration is 1.62 for measview2 of EVM.

## CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker amplitude for the selected marker in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:EVM[:MEASview<y>]:MARKer<x>:DELTA:Y?`

## Related Commands

[CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:DELTA:X?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in percent (%).

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:DELta:Y?` might return 1.62, indicating that the delta marker amplitude is 1.62% for measview2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the trace in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:EVM[:MEASview<y>]:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:MAXimum` moves Marker 1 (M1) to the highest peak on the trace for measview2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the EVM versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).



## Group

Calculate commands

## Syntax

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer](#)

## Arguments

None

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace for measview2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the trace in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:PEAK:LEFT` moves the Marker 1 (M1) to next peak to the left on the trace for measview2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the EVM versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time)

## Group

Calculate commands

## Syntax

`CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Related Commands

[\*CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer\*](#)

## Arguments

None

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace for measview2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the trace in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:PEAK:RIGHT` moves the Marker 1 (M1) to the next peak to the right on the trace for measview2 of EVM.

## CALCulate:EVM[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the EVM versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:EVM[:MEASview<y>]:MARKer<x>:X?
```

## Arguments

<value>::=<NRf> specifies the horizontal position of the marker Range: (analysis offset) to (analysis offset) + (analysis length). The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`CALCulate:EVM:MEASview2:MARKer1:X 1.5`  $\mu$ s places Marker 1 (M1) at 1.5  $\mu$ s on the trace for measview 2 of EVM.

## **CALCulate:EVM[:MEASview<y>]:MARKer<x>:Y? (Query only)**

Queries the marker amplitude of the selected marker in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:EVM[:MEASview<y>]:MARKer<x>:Y?`

## Related Commands

[\*CALCulate:EVM\[:MEASview<y>\]:MARKer<x>:X\*](#)

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in percent (%).

## Examples

`CALCulate:EVM:MEASview2:MARKer<x>:Y?` might return 5.34, indicating Marker 1 (M1) is at 5.34% for measview2 of EVM.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]? (Query only)**

Returns the delta marker time for the selected marker on the Frequency deviation versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:FDVTime:MEASview<y>:MARKer<x>:DELTA:X[:TIME]?`

## Related Commands

[CALCulate:FDVTime\[:MEASVIEW<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:DELTA:X?` might return 120.0E-9, indicating that the delta marker duration is 120ns.

## CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker amplitude for the selected marker on the Frequency deviation versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:FDVTime:MEASview<y>2:MARKer<x>:DELTA:Y?`

## Related Commands

[CALCulate:FDVTime:MEASVIEW2:MARKer<x>:DELTA:X\[:TIME\]?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in Hz.

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:DELTA:Y?` might return `27.05E+3`, indicating that the delta marker amplitude is 27.05KHz.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the trace in the Frequency deviation versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation vs Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:FDVTime:MEASview<y>:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the Frequency deviation versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency Deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:FDVTime\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer](#)

## Arguments

None

## Examples

CALCULATE:FDVTime:MEASview2:MARKER1:PEAK:HIGHER moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)

Moves the selected marker to the next peak to the left on the Frequency deviation versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:FDVTime:MEASview<y>:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:FDVTime\[:MEASVIEW<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the Frequency deviation vs Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation vs Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:FDVTime:MEASview<y>:MARKer<x>:PEAK:LOWer`

## Related Commands

[\*CALCulate:FDVTime\[:MEASVIEW<y>\]:MARKer<x>:PEAK:HIGHer\*](#)

## Arguments

None

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the trace in the Frequency deviation vs Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).



## Conditions

Measurement views: Frequency deviation vs Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:FDVTime:MEASview<y>:MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:FDVTime\[:MEASVIEW<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

CALCULATE:FDVTime:MEASview2:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:X[:TIME]

Sets or queries the horizontal position (time) of the selected marker in the Frequency deviation vs Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation vs Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:FDVTime:MEASview<y>:MARKer<x>:X[:TIME] <value>
```

```
CALCulate:FDVTime:MEASview<y>:MARKer<x>:X[:TIME] ?
```

## Related Commands

[CALCulate:FDVTime\[:MEASview<y>\]:MARKer<x>:Y? \(Query only\)](#)

## Arguments

None.

## Returns

<NRf> Marker amplitude of the selected marker in Hz.

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:X -15.34E+06` places Marker 1 (M1) at -15.34MHz on the trace.

## **CALCulate:FDVTime[:MEASview<y>]:MARKer<x>:Y? (Query only)**

Queries the marker amplitude of the selected marker in the Frequency deviation versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:FDVTime:MEASview<y>:MARKer<x>:Y?`

## Related Commands

[CALCulate:FDVTime\[:MEASVIEW<y>\]:MARKer<x>:X\[:TIME\]](#)

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in Hz.

## Examples

`CALCULATE:FDVTime:MEASview2:MARKER1:Y?` might return -15.34E+6, indicating Marker 1 (M1) is at -15.34 MHz.

## **CALCulate:{FSETtling|PSETtling}:MARKer<x>:DELTA:X? (Query only)**

Returns the delta marker time for the selected marker in the Frequency and Phase Settling displays.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Calculate commands

### Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:DELTA:X?
```

### Related Commands

```
CALCulate:FVTime:MARKer<x>:DELTA:Y?
```

### Arguments

None

### Returns

<NRf> Delta marker time for the selected marker.

### Examples

CALCULATE:FSETTLING:MARKER1:DELTA:X? might return 120.0E-9, indicating that the delta marker time is 120 ns.

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker frequency for the selected marker in the Frequency and Phase Settling displays.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Calculate commands

### Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:DELTA:Y?
```

### Related Commands

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:DELTA:X?
```

## Arguments

None

## Returns

<NRf> Delta marker frequency for the selected marker in Hz.

## Examples

`CALCULATE:FSETTLING:MARKER1:DELTA:Y?` might return `27.05E+3`, indicating that the delta marker frequency is 27.05 kHz.

## **CALCulate:{FSETtling|PSETtling}:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the trace in the Frequency or Phase Settling display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Calculate commands

## Syntax

`CALCulate:{FSETtling|PSETtling}:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCULATE:FSETTLING:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Frequency or Phase Settling trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Calculate commands

## Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:HIGHer
```

## Related Commands

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALCULATE:FSETTLING:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Frequency or Phase Settling display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Calculate commands

## Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCULATE:FSETTLING:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Frequency or Phase Settling trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Calculate commands

## Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:HIGHer
```

## Arguments

None

## Examples

```
CALCULATE:FSETTLING:MARKER1:PEAK:LOWER moves Marker 1 (M1) to the next peak lower in amplitude on the trace.
```

## **CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the Frequency or Phase Settling display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:FVTime:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

```
CALCULATE:FSETTLING:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.
```

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:TRACe

Sets or queries the trace on which the specified marker is placed in the spectrum measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Calculate commands

### Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:TRACe { TRACE1 |TRACE2 }
```

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:TRACe?
```

### Arguments

TRACE1 places the specified marker on Trace 1.

TRACE2 places the specified marker on Trace 2.

### Examples

```
CALCULATE:FSETTLING:MARKER1:TRACE TRACE1 places Marker 1 (M1) on Trace 1.
```

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Frequency and Phase Settling display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Calculate commands

### Syntax

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:X <value>
```

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:X?
```

### Related Commands

```
CALCulate:{FSETtling|PSETtling}:MARKer<x>:Y?
```

## Arguments

<value> : :=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

## Examples

CALCULATE:FVTIME:MARKER1:X 1.5u places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

## CALCulate:{FSETtling|PSETtling}:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Frequency and Phase Settling display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Calculate commands

## Syntax

CALCulate:{FSETtling|PSETtling}:MARKer<x>:Y?

## Related Commands

CALCulate:{FSETtling|PSETtling}:MARKer<x>:X

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in Hz or degrees.

## Examples

CALCULATE:FSETTLING:MARKER1:Y? might return -15.34E+6, indicating Marker 1 (M1) is at -15.34 MHz.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker time for the selected marker in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands



## Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Related Commands

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:DELTA:X? might return 120.0E-9, indicating that the delta marker time is 120 ns.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker frequency for the selected marker in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker frequency for the selected marker in Hz.

## Examples

`CALCULATE:FVTIME:MEASview1:MARKER1:DELTA:Y?` might return `27.05E+3`, indicating that the delta marker frequency is 27.05 kHz.

## **CALCulate:FVTime[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the trace in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Calculate commands

### Syntax

`CALCulate:FVTime[:MEASview<y>]:MARKer<x>:MAXimum`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value 1.

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Examples

`CALCULATE:FVTIME:MEASview1:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Frequency versus Time trace.

### Conditions

Measurement views: Frequency versus Time

### Group

Calculate commands

### Syntax

`CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

### Related Commands

`CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:PEAK:HIGHER moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Frequency versus Time trace.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:PEAK:LOWER moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:X?
```

### Related Commands

CALCulate:FVTime[:MEASview<y>]:MARKer<x>:Y?

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

### Examples

```
CALCULATE:FVTIME:MEASview1:MARKER1:X 1.5u places Marker 1 (M1) at 1.5 µs on the trace.
```

## CALCulate:FVTime[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:FVTime[:MEASview<y>]:MARKer<x>:Y?
```

### Related Commands

CALCulate:FVTime[:MEASview<y>]:MARKer<x>:X

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Marker amplitude of the selected marker in Hz.

## Examples

CALCULATE:FVTIME:MEASview1:MARKER1:Y? might return -15.34E+6, indicating Marker 1 (M1) is at -15.34MHz.

## CALCulate:HARMonic:APPLy (No Query Form)

Based on the fundamental frequency and harmonic factor values the harmonic markers are applied and markers are placed in their respective positions on the plot.

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

CALCulate:HARMonic:APPLy

## Arguments

None

## Returns

None

## Examples

CALCULATE:HARMONIC:APPLY places the harmonic markers in their respective positions.

## CALCulate:HARMonic:MARKer<x>:FACTor

Sets or queries the factor value for the specified harmonic marker in EMC view.

<x> = <NR1> 1 to 5 (marker number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Calculate commands

## Syntax

```
CALCulate:HARMonic:MARKer<x>:FACTor <value>
```

```
CALCulate:HARMonic:MARKer<x>:FACTor?
```

## Arguments

<value> =

Either

OEIG = 1/8

QUAR = 1/4

HALF = 1/2

ONE = 1

TWO = 2

THREE = 3

FOUR = 4

FIVE = 5

SIX = 6

SEVEN = 7

EIGHT = 8

NINE = 9

TEN = 10

## Returns

<String> See Arguments

## Examples

CALCulate:HARMonic:MARKer1:FACTor? might return 4 indicating that the factor value of harmonic marker 1 is 4.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker time for the selected marker in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:X?

## Related Commands

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:DELTA:X? might return 120.0E-9, indicating that the delta marker time is 120 ns.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

## Syntax

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?

## Related Commands

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:DELTA:X?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.



## Returns

<NRf> Delta marker amplitude for the selected marker in volts.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:DELTA:Y? might return -3.45E-3, indicating that the delta marker amplitude is -3.45 mV.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:MAXimum

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:MAXIMUM moves Marker 1 (M1) to the highest peak on the trace.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the RF I&Q versus Time trace.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer

## Related Commands

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:PEAK:HIGHER moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the RF I&Q versus Time trace.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCULATE:IQVTIME:MEASview1:MARKER1:PEAK:LOWER moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

`CALCULATE:IQVTIME:MEASview1:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:TRACe**

Places the selected marker on the I or Q trace in the RF I&Q versus Time measurement. The query command returns which trace the selected marker is placed on.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:TRACe { TRACE1 | TRACE2 }
```

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:TRACe?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

TRACE1 places the selected marker on the I trace.

TRACE2 places the selected marker on the Q trace.

## Examples

`CALCULATE:IQVTIME:MEASview1:MARKER1:TRACE` TRACE1 places Marker 1 (M1) on the I trace.

## **CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker in the RF I&Q versus Time measurement.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Calculate commands

### Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:X?
```

## Related Commands

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

## Examples

```
CALCULATE:IQVTIME:MEASview1:MARKER1:X 1.5us places Marker 1 (M1) at 1.5 µs on the trace.
```

## CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:IQVTime[:MEASview<y>]:MARKer<x>:X
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Marker amplitude of the selected marker in volts.

## Examples

```
CALCULATE:IQVTIME:MEASview1:MARKER1:Y? might return 25.803E-3, indicating Marker 1 (M1) is at 25.803 mV.
```

## **CALCulate:LTE:ACLR:MARKer:<x>PEAK:HIGHer (No Query Form)**

Moves the specified marker to the next peak higher in amplitude on the trace in the LTE ACLR display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement view: LTE ACLR

### **Group**

Calculate commands

### **Syntax**

CALCulate:LTE:ACLR:MARKer:<x>PEAK:HIGHer

### **Arguments**

None

### **Examples**

CALCULATE:LTE:ACLR:MARKER1:PEAK:HIGHER moves marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:LTE:ACLR:MARKer<x>:DELTA:X? (Query Only)**

Returns the frequency of the specified delta marker on the trace in the LTE ACLR display. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement view: LTE ACLR

<x> parameters: 1 to 4; MARKer0 (reference marker) is not valid.

### **Group**

Calculate commands

### **Syntax**

CALCulate:LTE:ACLR:MARKer<x>:DELTA:X?

### **Returns**

<NR3>

### **Examples**

CALCULATE:LTE:ACLR:MARKER1:DELTA:X? might return 1.2500000E+6, indicating that the frequency of the specified delta marker is 1.25 MHz.

## CALCulate:LTE:ACLR:MARKer<x>:DELTA:Y? (Query Only)

Returns the amplitude for the specified delta marker on the trace in the LTE ACLR display. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement view: LTE ACLR

<x> parameters: 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

CALCulate:LTE:ACLR:MARKer<x>:DELTA:Y?

### Returns

<NRf>

### Examples

CALC:LTEACLR:MARK1:DELTA:Y? might return 23.45, indicating that the amplitude of the specified delta marker is 23.45 dB.

## CALCulate:LTE:ACLR:MARKer<x>:MAXimum (No Query Form)

Positions the specified marker at the highest peak on the trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

<x> parameters: 0 to 4

### Group

Calculate commands

### Syntax

CALCulate:LTE:ACLR:MARKer<x>:MAXimum

### Arguments

None

### Examples

CALCULATE:LTE:ACLR:MARKER1:MAXIMUM moves marker 1 (M1) to the highest peak on the trace.

## **CALCulate:LTE:ACLR:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

<x> parameters: 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:LTE:ACLR:MARKer<x>:PEAK:LEFT
```

### **Arguments**

None

### **Examples**

`CALCULATE:LTE:ACLR:MARKER1:PEAK:LEFT` moves marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:LTE:ACLR:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the trace in the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

<x> parameters: 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:LTE:ACLR:MARKer<x>:PEAK:LOWer
```

### **Arguments**

None

### **Examples**

`CALCULATE:LTE:ACLR:MARKER1:PEAK:LOWER` moves marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:LTE:ACLR:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the LTE ACLR display.



## Conditions

Measurement view: LTE ACLR

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:ACLR:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

CALCULATE:LTE:ACLR:MARKER1:PEAK:RIGHT moves marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:LTE:ACLR:MARKer<x>:SET:CENTer (No Query Form)

Moves the specified marker to the center frequency in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:ACLR:MARKer<x>:SET:CENTer
```

## Arguments

None

## Examples

CALCULATE:LTE:ACLR:MARKER1:SET:CENTER moves marker 1 (M1) to the center frequency on the trace.

## CALCulate:LTE:ACLR:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the trace in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:ACLR:MARKer<x>:X <value>
```

```
CALCulate:LTE:ACLR:MARKer<x>:X?
```

## Arguments

<value> : :=<NRf> specifies the horizontal position of the marker.

## Returns

<NRf>

## Examples

```
CALCULATE:LTE:ACLR:MARKER1:X 800.00000000E+6 places marker 1 (M1) at 800 MHz on the trace.
```

## CALCulate:LTE:ACLR:MARKer<x>:Y? (Query Only)

Queries the amplitude of the selected marker on the trace in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:ACLR:MARKer<x>:Y?
```

## Returns

<NRf>

## Examples

```
CALCULATE:LTE:ACLR:MARKER1:Y? might return -34.28, indicating that the amplitude of marker 1 (M1) is -34.28 dBm.
```

## CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:X? (Query Only)

Returns the frequency of the specified delta marker on the trace in the LTE Channel Spectrum display. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 1 to 4; MARKer0 (reference marker) is not valid.

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:X?
```

## Returns

<NR3>

## Examples

CALCULATE:LTE:CHSPECTRUM:MARKER1:DELTA:X? might return 1.2500000E+6, indicating that the frequency of the specified delta marker is 1.25 MHz.

## CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:Y? (Query Only)

Returns the amplitude for the specified delta marker on the trace in the LTE Channel Spectrum display. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 1 to 4; MARKer0 (reference marker) is not valid.

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:DELTA:Y?
```

## Returns

<NRf>

## Examples

CALC:LTE:CHSPECTRUM:MARK1:DELTA:Y? might return 23.45, indicating that the amplitude of the specified delta marker is 23.45 dB.

## CALCulate:LTE:CHSPectrum:MARKer<x>:MAXimum (No Query Form)

Positions the marker at the highest peak on the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCULATE:LTE:CHSPECTRUM:MARKER1:MAXIMUM` moves marker 1 (M1) to the highest peak on the trace.

## **CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCULATE:LTE:CHSPECTRUM:MARKER1:PEAK:LEFT` moves marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

CALCULATE:LTE:CHSPECTRUM:MARKER1:PEAK:RIGHT moves marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:LTE:CHSPectrum:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:X <value>
```

```
CALCulate:LTE:CHSPectrum:MARKer<x>:X?
```

## Arguments

<value>::=<NRf> specifies the horizontal position of the marker.

## Returns

<NRf>

## Examples

CALCULATE:LTE:CHSPECTRUM:MARKER1:X 800.000000000E+6 places marker 1 (M1) at 800 MHz on the trace.

## CALCulate:LTE:CHSPectrum:MARKer<x>:Y? (Query Only)

Queries the amplitude of the selected marker on the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CHSPectrum:MARKer<x>:Y?
```

## Returns

<NRf>

## Examples

CALCULATE:LTE:CHSPECTRUM:MARKER1:Y? might return -34.28, indicating that the amplitude of marker 1 (M1) is -34.28 dBm.

## CALCulate:LTE:CONSte:MARKer<x>:FREQuency

Sets or returns the subcarrier number of the LTE symbol in the LTE Constellation display. This constellation shows only primary and secondary synchronization signals. Any input outside this range will show the marker set to -31 or +31.

## Conditions

Measurement view: LTE Constellation

<x> parameters: 0 to 4

Range: -31 to +31

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:CONSte:MARKer<x>:FREQuency <value>
```

```
CALCulate:LTE:CONSte:MARKer<x>:FREQuency?
```

## Arguments

<value> : : <NRf> specifies the subcarrier number of the specified marker

## Returns

<NRf> The value type depends on where the marker is placed.

## Examples

`CALCULATE:LTE:CONSTE:MARKER2:FREQUENCY?` might return +31, indicating that Marker 2 (M2) is positioned at the subcarrier +31 in the LTE Constellation display.

## CALCulate:LTE:CONSte:MARKer<x>:MAGNitude? (Query Only)

Queries the marker magnitude of the selected marker in the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

<x> parameters: 0 to 4

### Group

Calculate commands

### Syntax

`CALCulate:LTE:CONSte:MARKer<x>:MAGNitude?`

### Returns

<NRf> The magnitude for the selected marker.

## Examples

`CALCULATE:LTE:CONSTE:MARKER2:MAGNITUDE` might return "1.004" indicating readout of Marker 2 (M2) is "1.004".

## CALCulate:LTE:CONSte:MARKer<x>:PHASe? (Query Only)

Queries the phase readout of the selected marker in the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Calculate commands

### Syntax

`CALCulate:LTE:CONSte:MARKer<x>:PHASe?`

### Returns

<NRf> The phase readout for the selected marker.

## Examples

`CALCULATE:LTE:CONSTE:MARKER1:PHASE?` might return 35.74, indicating the phase readout of Marker 1 (M1) is 35.74°

## **CALCulate:LTE:CONSte:MARKer<x>:TIME**

Sets or queries the X position (Symbol Value) of the selected marker in the LTE Constellation display.

### **Conditions**

Measurement view: LTE Constellation

<x> parameters: 0 to 4

For FDD, the range is 0 and 1.

For TDD the range is 0 and 3.

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:LTE:CONSte:MARKer<x>:TIME <value>
```

```
CALCulate:LTE:CONSte:MARKer<x>:TIME?
```

### **Arguments**

<value> ::= <NRf> specifies the specified marker symbol number.

### **Returns**

<NR1>

### **Examples**

CALCULATE:LTE:CONStE:MARKER<x>:TIME 0 places Marker 1 (M1) at the symbol 0 on range for time.

## **CALCulate:LTE:CONSte:MARKer<x>:TYPE? (Query Only)**

Queries the data type of the selected marker in LTE Constellation display. The data type can either be “PSS” (Primary Synchronization Signal) or “SSS” (Secondary Synchronization Signal).

### **Conditions**

Measurement view: LTE Constellation

<x> parameters: 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:LTE:CONSte:MARKer<x>:TYPE?
```



## Returns

PSS means that the data type for the selected marker is Primary Synchronization Signal

SSS means that the data type for the selected marker is Secondary Synchronization Signal

## Examples

`CALCULATE:LTE:CONSTE:MARKER1:TYPE?` might return SSS, indicating that the data type for the Marker 1 (M1) is SSS.

## CALCulate:LTE:PVTime:MARKer:<x>PEAK:HIGHer (No Query Form)

Moves the specified marker to the next peak higher in amplitude on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

Parameter <x>: 0 to 4

## Group

Calculate commands

## Syntax

`CALCulate:LTE:PVTime:MARKer:<x>PEAK:HIGHer`

## Arguments

None

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:PEAK:HIGHER` moves marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:LTE:PVTime:MARKer<x>:DELTA:X? (Query Only)

Returns the frequency of the specified delta marker on the trace in the LTE Power vs Time display. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 1, 2, 3...(valid integral values range from 1 to the number of markers active for the display); MARKer0 (reference marker) is not valid.

Querying an out-of-range value causes an execution error (-222, "Data out of range").

## Group

Calculate commands

## Syntax

`CALCulate:LTE:PVTime:MARKer<x>:DELTA:X?`

## Returns

<NR3>

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:DELTA:X?` might return `1.2800000E-6`, indicating that the frequency of the specified delta marker is 1.28  $\mu$ s.

## **CALCulate:LTE:PVTime:MARKer<x>:DELTA:Y? (Query Only)**

Returns the amplitude for the specified delta marker on the trace in the LTE Power vs Time display using the currently selected amplitude unit. You can set/query the amplitude unit with the `SENSe:POWer:UNITs` command. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 1, 2, 3...(valid integral values range from 1 to the number of markers active for the display); MARKer0 (reference marker) is not valid.

Querying an out-of-range value causes an execution error (-222, "Data out of range").

## Group

Calculate commands

## Syntax

`CALCulate:LTE:PVTime:MARKer<x>:DELTA:Y?`

## Returns

<NRf>

## Examples

`CALCULATE:LTE:PVTIME:MARKER3:DELTA:Y?` might return `-12.91`, indicating that the amplitude of the delta marker 3 (M3) is `-12.91` dB.

## **CALCulate:LTE:PVTime:MARKer<x>:MAXimum (No Query Form)**

Positions the specified marker at the highest peak on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:PVTime:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:MAXIMUM` moves marker 1 (M1) to the highest peak on the trace.

## **CALCulate:LTE:PVTime:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:PVTime:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:PEAK:LEFT` moves marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:LTE:PVTime:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

`CALCulate:LTE:PVTime:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:PEAK:LOWER` moves marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:LTE:PVTime:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4

## Group

Calculate commands

## Syntax

`CALCulate:LTE:PVTime:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Examples

`CALCULATE:LTE:PVTIME:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace

## **CALCulate:LTE:PVTime:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker on the trace in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4

Range: Start to Stop time (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:PVTime:MARKer<x>:X <NRf>
```

```
CALCulate:LTE:PVTime:MARKer<x>:X?
```

## Arguments

<NRf> specifies the value of the horizontal position of the marker.

## Returns

<NRf>

## Examples

```
CALCULATE:LTE:PVTIME:MARKER3:X 1.000000000E-6 places Marker 3(M3) at 1  $\mu$ s on the trace.
```

## CALCulate:LTE:PVTime:MARKer<x>:Y? (Query Only)

Queries the amplitude of the selected marker on the trace in the LTE Power vs Time display. You can set/query the amplitude units with the `SENSe:POWer:UNITs` command.

## Conditions

Measurement view: LTE Power vs Time

<x> parameters: 0 to 4 Querying an out-of-range value causes an execution error (-222, "Data out of range").

## Group

Calculate commands

## Syntax

```
CALCulate:LTE:PVTime:MARKer<x>:Y?
```

## Returns

<NRf> specifies the value of the horizontal position of the marker.

## Examples

```
CALCULATE:LTE:PVTIME:MARKER1:Y? might return 33.393856E-3, indicating that the amplitude of marker 1 (M1) is 0.033 dBm.
```

## CALCulate:MARKer:ADD (No Query Form)

Adds a marker. Every execution of this command adds a marker from MR, then M1 to M4, sequentially.



**Note:** If all markers are already turned on, the error message "Cannot add another marker" (execution error -200) is returned.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:ADD
```

## Related Commands

CALCulate:MARKer:AOff, CALCulate:MARKer:DElete

## Arguments

None

## Examples

CALCULATE : MARKER : ADD adds a marker.

## CALCulate:MARKer:AOff (No Query Form)

Turns off all markers.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:AOff
```

## Related Commands

CALCulate:MARKer:ADD, CALCulate:MARKer:DElete

## Arguments

None

## Examples

CALCULATE : MARKER : AOff turns off all markers.

## CALCulate:MARKer:DElete (No Query Form)

Deletes the last marker added.



**Note:** *If all markers are turned off, the error message "Cannot delete another marker" (execution error -200) is returned.*

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

`CALCulate:MARKer:DELeTe`

## Related Commands

`CALCulate:MARKer:ADD`, `CALCulate:MARKer:AOff`

## Arguments

None

## Examples

`CALCULATE:MARKER:DELETE` deletes the last marker added.

## CALCulate:MARKer:DENSity:EXCursion

Sets or queries the minimum excursion of DPX signal density, or how far the density (hit count for pixels) must be above the surrounding noise to be detected as a peak. This command is effective for the DPX bitmap trace.

For marker peak up and marker peak down operations, the signal is considered to be a peak if it exceeds the excursion and the threshold set by the `CALCulate:MARKer:DENSity:THReshold` command.

For marker peak left and marker peak right operations, the signal is considered to be a peak if it first exceeds the amplitude excursion set by `CALCulate:MARKer:PEAK:EXCursion`, the amplitude threshold set by `CALCulate:MARKer:PEAK:THReshold`, the density excursion set by `CALCulate:MARKer:DENSity:EXCursion` and the density threshold set by `CALCulate:MARKer:DENSity:THReshold`.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

`CALCulate:MARKer:DENSity:EXCursion <number>`

`CALCulate:MARKer:DENSity:EXCursion?`

## Related Commands

`CALCulate:MARKer:DENSity:THReshold`, `CALCulate:MARKer:PEAK:EXCursion`, `CALCulate:MARKer:PEAK:THReshold`

## Arguments

`<number> ::= <NR1>` specifies the minimum excursion density. Range: 0 to 100%.

## Examples

`CALCULATE:MARKER:DENSITY:EXCURSION30` sets the minimum excursion density to 30.

## CALCulate:MARKer:DENSity:SMOothing

Sets or queries the number of pixels squared for smoothing the DPX signal density. This command is effective for the DPX bitmap trace.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

`CALCulate:MARKer:DENSity:SMOothing <number>`

`CALCulate:MARKer:DENSity:SMOothing?`

## Arguments

`<number> ::= <NR1>` specifies the number of pixels squared for smoothing the DPX signal density. Range: 1 to 20.

## Examples

`CALCULATE:MARKER:DENSITY:SMOOTHING5` sets the smoothing number to 5.

## CALCulate:MARKer:DENSity:THReshold

Sets or queries the threshold of DPX signal density above which the density (hit count for pixels) is detected as a peak. This command is effective for the DPX bitmap trace.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

`CALCulate:MARKer:DENSity:THReshold <number>`

`CALCulate:MARKer:DENSity:THReshold?`



## Related Commands

CALCulate:MARKer:DENSity:EXCursion

## Arguments

<number>:::=<NR1> specifies the threshold density for detecting peaks. Range: 0 to 100%.

## Examples

CALCULATE:MARKER:DENSITY:THRESHOLD

THRESHOLD300 sets the threshold density to 300.

## CALCulate:MARKer:DRAG:SEARch:STATe

Enables or disables peak searching when dragging a marker in a plot.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:DRAG:SEARch:STATe { OFF | ON | 0 | 1 }
```

```
CALCulate:MARKer:DRAG:SEARch:STATe?
```

## Arguments

OFF or 0 disables peak searching in a plot.

ON or 1 enables peak searching in a plot.

## Examples

CALCULATE:MARKER:DRAG:SEARCH:STATE OFF disables peak searching when dragging a marker in a plot.

## CALCulate:MARKer:MODE

Sets or queries the marker mode.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:MODE { ABSolute | DELTa }
```

```
CALCulate:MARKer:MODE?
```

## Arguments

`ABSolute` selects the absolute marker mode, in which the marker readout indicates the absolute value. `DELTA` selects the delta marker mode, in which the marker readout indicates the relative value to the reference marker.

## Examples

`CALCULATE:MARKER:MODEDELTA` selects the delta marker mode.

## CALCulate:MARKer:MODE

Sets or queries the marker mode.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:MODE { ABSolute | DELTa | POWer }
```

```
CALCulate:MARKer:MODE?
```

## Arguments

`ABSolute` selects the absolute marker mode, in which the marker readout indicates the absolute value.

`DELTA` selects the delta marker mode, in which the marker readout indicates the relative value to the selected marker.

`POWER` selects the power marker mode, in which the marker readout indicates the power value of the selected marker.

## Examples

`CALCULATE:MARKER:MODE POWER` selects the power marker mode.

`CALCULATE:MARKER:MODE ?` might return `POW`, meaning the Power is the assigned mode for the marker readout.

## CALCulate:MARKer:NOISe:MODE

Sets or queries the view/hide status of the dBc/Hz Marker noise reading.

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:NOISe:MODE { 0 | 1 }
```

## Arguments

0 disables the dBc/Hz reading.

1 enables the dBc/Hz reading.

## Returns

0 means the dBc/Hz reading is not available.

1 means the dBc/Hz reading is not available.

## Examples

CALCULATE:MARKER:NOISE:MODE 0 will disable the dBc/Hz marker noise reading.

## CALCulate:MARKer:PEAK:EXCursion

Sets or queries the minimum excursion level (how far a signal must be above the surrounding noise to be detected as a peak). The signal is considered to be a peak if it exceeds the minimum excursion level and the threshold level set by the CALCulate:MARKer:PEAK:THReshold command.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:PEAK:EXCursion <value>
```

```
CALCulate:MARKer:PEAK:EXCursion?
```

## Arguments

<value>::=<NRf> specifies the excursion level. Range: 0 to 100 dB.

## Examples

CALCULATE:MARKER:PEAK:EXCURSION10 sets the excursion level to 10 dB.

## CALCulate:MARKer:PEAK:THReshold

Sets or queries the threshold level above which a signal is detected as a peak.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:MARKer:PEAK:THReshold <value>
```

```
CALCulate:MARKer:PEAK:THReshold?
```

## Arguments

<value>: :=<NRf> specifies the threshold level for detecting peaks. Range: -170 to +130 dBm.

## Examples

```
CALCULATE:MARKER:PEAK:THRESHOLD-50 sets the threshold level to -50 dBm.
```

## CALCulate:MCPower:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the selected marker in the MCPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: MCPR

## Group

Calculate commands

## Syntax

```
CALCulate:MCPower:MARKer<x>:DELTA:X?
```

## Related Commands

```
CALCulate:MCPower:MARKer<x>:DELTA:Y?
```

## Arguments

None

## Returns

<NRf> Delta marker frequency for the selected marker.

## Examples

```
CALCULATE:MCPOWER:MARKER1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.
```

## CALCulate:MCPower:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the MCPR measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

```
CALCulate:MCPower:MARKer<x>:DELTA:Y?
```

### Related Commands

```
CALCulate:MCPower:MARKer<x>:DELTA:X?
```

### Arguments

None

### Returns

<NRf> Delta marker amplitude for the selected marker.

### Examples

```
CALCULATE:MCPOWER:MARKER1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.
```

## CALCulate:MCPower:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

```
CALCulate:MCPower:MARKer<x>:MAXimum
```

### Arguments

None

## Examples

`CALCULATE:MCPOWER:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:MCPower:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

`CALCulate:MCPower:MARKer<x>:PEAK:LEFT`

### Related Commands

`CALCulate:MCPower:MARKer<x>:PEAK:RIGHT`

### Arguments

None

## Examples

`CALCULATE:MCPOWER:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:MCPower:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

`CALCulate:MCPower:MARKer<x>:PEAK:RIGHT`

### Related Commands

`CALCulate:MCPower:MARKer<x>:PEAK:LEFT`

### Arguments

None

## Examples

`CALCULATE:MCPOWER:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:MCPower:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

`CALCulate:MCPower:MARKer<x>:X <value>`

`CALCulate:MCPower:MARKer<x>:X?`

### Related Commands

`CALCulate:MCPower:MARKer<x>:Y?`

### Arguments

`<value>::=<NRf>` specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

`CALCULATE:MCPOWER:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

## **CALCulate:MCPower:MARKer<x>:Y? (Query Only)**

Queries the marker amplitude of the selected marker in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Calculate commands

### Syntax

`CALCulate:MCPower:MARKer<x>:Y?`

### Related Commands

`CALCulate:MCPower:MARKer<x>:X`

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

`CALCULATE:MCPOWER:MARKER1:Y?` might return `-34.28`, indicating Marker 1 (M1) is at `-34.28` dBm.

## **CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:X? (Query only)**

Returns the delta marker time for the selected marker in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:X?`

## Related Commands

[CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in percent (%).

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:DELTA:X?` might return `3.84`, indicating that the delta marker duration is `3.84`.



## CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker amplitude for the selected marker in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

### Related Commands

[CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:DELTA:X?](#)

### Arguments

None

### Returns

<NRf> Delta marker amplitude for the selected marker in percent (%).

### Examples

CALCulate:MERRor:MEASview2:MARKer1:DELTA:Y? might return 3.84, indicating that the delta marker amplitude is 3.84%.

## CALCulate:MERRor[:MEASview<y>]:MARKer<x>:MAXimum (No query form)

Moves the selected marker to the highest peak on the trace in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the Magnitude error versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[\*CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer\*](#)

## Arguments

None

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the trace in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

### **Related Commands**

[CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

### **Arguments**

None

### **Examples**

CALCulate:MERRor:MEASview2:MARKer1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the Magnitude error versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

### **Group**

Calculate commands

## Syntax

`CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Related Commands

[\*CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer\*](#)

## Arguments

None

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the trace in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Calculate commands

## Syntax

`CALCulate:MERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT`

## Related Commands

[\*CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT\*](#)

## Arguments

None

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:MERRor[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Magnitude error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:X?
```

### Arguments

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

### Examples

```
CALCulate:MERRor:MEASview2:MARKer1:X 1.5µs places Marker 1 (M1) at 1.5 µs on the trace.
```

## CALCulate:MERRor[:MEASview<y>]:MARKer<x>:Y? (Query only)

Queries the marker amplitude of the selected marker in the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:MERRor[:MEASview<y>]:MARKer<x>:Y?
```

## Related Commands

[CALCulate:MERRor\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in percent (%).

## Examples

`CALCulate:MERRor:MEASview2:MARKer1:Y?` might return 5.34, indicating Marker 1 (M1) is at 5.34%.

## CALCulate:NOISe:FIGure:MARKer<x>:DELTA:X? (Query Only)

Queries the frequency difference between MR (reference marker) and the selected marker.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

## Group

Calculate commands

## Syntax

`CALCulate:NOISe:FIGure:MARKer<x>:DELTA:X?`

## Related Commands

[CALCulate:NOISe:FIGure:MARKer<x>:Y?](#)

## Returns

<NRf> is the frequency difference value between MR and the selected marker.

## Examples

`CALCulate:NOISe:FIGure:MARKer1:DELTA:X?` might return 0.0000, indicating the frequency difference between MR (reference marker) and the selected marker (M1) is 0.0000 dB.

## CALCulate:NOISe:FIGure:MARKer<x>:DELTA:Y? (Query only)

Queries the power difference value between MR (reference marker) and the selected marker.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:NOISe:FIGure:MARKer<x>:X?
```

## Returns

<NRf> is the power difference value between MR and the selected marker.

## Examples

`CALCULATE:NOISE:FIGURE:MARKER1:DELTA:Y?` might return 18.7647705078, indicating the difference between MR (reference marker) and the selected marker (M1) is 18.76 dB.

## CALCulate:NOISe:FIGure:MARKer<x>:MAXimum (No Query Form)

Places the selected marker at the maximum value on the peak of the trace.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:MAXimum
```

## Related Commands

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer
```

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer
```

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT
```

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Returns

None

## CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer (No Query Form)

Places the selected marker at the next highest peak on the trace.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer
```

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Returns

None

## CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT (No Query Form)

Places the selected marker at the next peak on the trace to the left of the selected marker.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT
```



## Related Commands

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT

CALCulate:NOISe:FIGure:MARKer<x>:MAXimum

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer

## Arguments

None

## Returns

None

## CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer (No Query Form)

Places the selected marker at the next lowest peak on the trace.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer
```

## Related Commands

CALCulate:NOISe:FIGure:MARKer<x>:MAXimum

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Returns

None

## CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT (No Query Form)

This command places the specified marker on a specified trace. The query form returns the trace on which the marker resides.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT
```

## Related Commands

CALCulate:NOISe:FIGure:MARKer<x>:MAXimum

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer

## Arguments

None

## Returns

None

## CALCulate:NOISe:FIGure:MARKer<x>:TRACe

Sets or queries which trace the specified marker is attached to.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:FIGure:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3 }
```

```
CALCulate:NOISe:FIGure:MARKer<x>:TRACe?
```

## Related Commands

CALCulate:NOISe:FIGure:MARKer<x>:MAXimum

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:HIGHer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LEFT

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:LOWer

CALCulate:NOISe:FIGure:MARKer<x>:PEAK:RIGHT

## Arguments

TRACE1 moves the specified marker to trace 1.

TRACE2 moves the specified marker to trace 2.

TRACE3 moves the specified marker to trace 3.

## Returns

TRACE1 means the specified marker is on trace 1.

TRACE2 means the specified marker is on trace 2.

TRACE3 means the specified marker is on trace 3.

## Examples

CALCULATE:NOISE:FIGURE:MARKER1:TRACE? might return TRACE2, indicating that M1 (marker 1) is on trace 2.

## CALCulate:NOISe:FIGure:MARKer<x>:X? (Query Only)

Queries the frequency of the specified marker.

## Conditions

Measurement view: Noise Figure

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

CALCulate:NOISe:FIGure:MARKer<x>:X?

## Related Commands

CALCulate:NOISe:FIGure:MARKer<x>:Y?

## Returns

<NR3> = a floating point value with an exponent. This is the frequency of the specified marker

## Examples

CALCULATE:NOISE:FIGURE:MARKER1:X? might return 608.000000000E+6, indicating the frequency of M1 (marker 1) is 608 MHz.

## **CALCulate:NOISe:FIGure:MARKer<x>:Y? (Query Only)**

Queries the vertical position of the specified marker.

### **Conditions**

Measurement view: Noise Figure

The parameter <x> = 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:NOISe:FIGure:MARKer<x>:Y?
```

### **Related Commands**

```
CALCulate:NOISe:FIGure:MARKer<x>:X?
```

### **Returns**

<NR2> = a floating point value without an exponent. This is the vertical position(dB) of the specified marker.

### **Examples**

CALCULATE:NOISE:FIGURE:MARKER1:Y? might return 18.7647705078, indicating the value of M1 (marker 1) is 18.76 dB.

## **CALCulate:NOISe:GAIN:MARKer<x>:DELTA:X? (Query Only)**

Query returns the frequency difference between MR (reference marker) and the selected marker.

### **Conditions**

Measurement view: Gain

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:NOISe:GAIN:MARKer<x>:DELTA:X?
```

### **Related Commands**

```
CALCulate:NOISe:GAIN:MARKer<x>:DELTA:Y?
```

### **Returns**

<NR2> = a floating point value without an exponent. This is the frequency difference value between MR and the selected marker.

## Examples

`CALCULATE:NOISE:GAIN:MARKER1:DELTA:X?` might return 100.0000, indicating the frequency difference between MR (reference marker) and the selected marker (M1) is 100 dB.

## **CALCulate:NOISe:GAIN:MARKer<x>:DELTA:Y? (Query Only)**

Query returns the power difference value between MR (reference marker) and the selected marker.

### Conditions

Measurement view: Gain

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

`CALCulate:NOISe:GAIN:MARKer<x>:DELTA:Y?`

### Related Commands

`CALCulate:NOISe:GAIN:MARKer<x>:DELTA:X?`

### Returns

<NR2> = a floating point value without an exponent. This is the power difference value between MR and the selected marker.

## Examples

`CALCULATE:NOISE:GAIN:MARKER1:DELTA:Y?` might return 18.7647705078, indicating the difference between MR (reference marker) and the selected marker(M1) is 18.76 dB.

## **CALCulate:NOISe:GAIN:MARKer<x>:MAXimum? (Query Only)**

Places the selected marker at the maximum value on the peak of the trace.

### Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

### Group

Calculate commands

### Syntax

`CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?`

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer? (Query Only)

Places the selected marker at the next highest peak on the trace.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT? (Query Only)

Places the selected marker at the next peak on the trace to the left of the selected marker.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT?
```

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer? (Query Only)

Places the selected marker at the next lowest peak on the trace.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LOWer?
```

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT? (Query Only)

Places the specified marker on a specified trace. The query form returns the trace on which the marker resides.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:GAIN:MARKer<x>:PEAK:RIGHT?
```

## Related Commands

CALCulate:NOISe:GAIN:MARKer<x>:MAXimum?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:GAIN:MARKer<x>:PEAK:HIGHer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:GAIN:MARKer<x>:TRACe

Sets or queries which trace the specified marker is attached to.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands



## Syntax

```
CALCulate:NOISe:GAIN:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3 }
```

```
CALCulate:NOISe:GAIN:MARKer<x>:TRACe?
```

## Arguments

TRACE1 moves the specified marker to trace 1.

TRACE2 moves the specified marker to trace 2.

TRACE3 moves the specified marker to trace 3.

## Returns

TRACE1 means the specified marker is on trace 1.

TRACE2 means the specified marker is on trace 2.

TRACE3 means the specified marker is on trace 3.

## Examples

CALCULATE:NOISE:GAIN:MARKER1:TRACE? might return TRACE2, indicating thatM1 (marker 1) is on trace 2.

## CALCulate:NOISe:GAIN:MARKer<x>:X? (Query Only)

This command queries the frequency of the specified marker.

## Conditions

Measurement view: Gain

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:GAIN:MARKer<x>:X?
```

## Related Commands

```
CALCulate:NOISe:GAIN:MARKer<x>:Y?
```

## Returns

<NR3>= a floating point value with an exponent. This is the frequency of the specified marker.

## Examples

CALCULATE:NOISE:GAIN:MARKER1:X? might return 608.000000000E+6, indicating the frequency of M1 (marker 1) is 608 MHz.

## **CALCulate:NOISe:GAIN:MARKer<x>:Y? (Query Only)**

Queries the vertical position (dB) of the specified marker.

### **Conditions**

Measurement view: Gain

The parameter <x> = 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:NOISe:GAIN:MARKer<x>:Y?
```

### **Related Commands**

```
CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:X?
```

### **Returns**

<NR2> = a floating point value without an exponent. This is the vertical position(dB) of the specified marker.

### **Examples**

CALCULATE:NOISE:GAIN:MARKER1:Y? might return 18.7647705078, indicating the value of M1 (marker 1) is 18.76 dB.

## **CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:X? (Query Only)**

Returns the frequency difference between MR (reference marker) and the selected marker.

### **Conditions**

Measurement view: Noise Temperature

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:X?
```

### **Related Commands**

```
CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:Y?
```

### **Returns**

<NR2> = a floating point value without an exponent. This is the frequency difference value between MR and the selected marker.

## Examples

`CALC:NOIS:TEMP:MARK1:DELTA:X?` might return 100.0000, indicating the frequency difference between MR (reference marker) and the selected marker(M1) is 100 dB.

## CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:Y? (Query Only)

Returns the temperature difference between MR (reference marker) and the selected marker.

### Conditions

Measurement view: Noise Temperature

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

`CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:Y?`

### Related Commands

`CALCulate:NOISe:TEMPerature:MARKer<x>:DELTA:X?`

### Returns

<NR3> = a floating point value with an exponent. This is the temperature difference value between MR and the selected marker.

## Examples

`CALC:NOIS:TEMP:MARK1:DELTA:Y?` might return 2.3663560181E+3, indicating the difference between MR (reference marker) and the selected marker(M1) is 2.36 kK.

## CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum? (Query Only)

Places the selected marker at the maximum value on the peak of the trace.

### Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

### Group

Calculate commands

### Syntax

`CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?`

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer? (Query Only)

Places the selected marker at the highest peak on the trace.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT? (Query Only)

Places the selected marker at the next peak on the trace to the left of the selected marker.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?
```

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer? (Query Only)

Places the selected marker at the next lowest peak on the trace.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?
```

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT? (Query Only)

This command places the specified marker on a specified trace. The query form returns the trace on which the marker resides.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:RIGHT?
```

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:MAXimum?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:TEMPerature:MARKer<x>:PEAK:LEFT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:TEMPerature:MARKer<x>:TRACe

Sets or queries which trace the specified marker is attached to.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

CALCulate:NOISe:TEMPerature:MARKer<x>:TRACe { TRACE1 | TRACE2| TRACE3 }

CALCulate:NOISe:TEMPerature:MARKer<x>:TRACe?

## Arguments

TRACE1 moves the specified marker to trace 1.

TRACE2 moves the specified marker to trace 2.

TRACE3 moves the specified marker to trace 3.

## Returns

TRACE1 means the specified marker is on trace 1.

TRACE2 means the specified marker is on trace 2.

TRACE3 means the specified marker is on trace 3.

## Examples

CALC:NOIS:TEMP:MARK1:TRAC? might return TRACE2, indicating that M1(marker 1) is on trace 2.

## CALCulate:NOISe:TEMPerature:MARKer<x>:X? (Query Only)

Queries the frequency of the specified marker.

## Conditions

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

CALCulate:NOISe:TEMPerature:MARKer<x>:X?

## Related Commands

CALCulate:NOISe:TEMPerature:MARKer<x>:Y?

## Returns

<NR2> = a floating point value without an exponent. This is the frequency of the specified marker.

## Examples

CALC:NOIS:TEMP:MARK1:X? might return 608.0000000000E+6, indicating the frequency of M1 (marker 1) is 608 MHz.

## **CALCulate:NOISe:TEMPerature:MARKer<x>:Y? (Query Only)**

Queries the temperature (Kelvin) at the specified marker.

### **Conditions**

Measurement view: Noise Temperature

The parameter <x> = 0 to 4

### **Group**

Calculate commands

### **Syntax**

CALCulate:NOISe:TEMPerature:MARKer<x>:Y?

### **Related Commands**

CALCulate:NOISe:TEMPerature:MARKer<x>:X?

### **Returns**

<NR3> = a floating point value with an exponent. This is the temperature at the specified marker.

### **Examples**

CALC:NOIS:TEMP:MARK1:Y? might return 1.4544410400E+3, indicating the value at M1 (marker 1) is 1.45 kK.

## **CALCulate:NOISe:YFACTOR:MARKer<x>:DELTA:X? (Query Only)**

Queries the frequency difference between MR (reference marker) and the selected marker.

### **Conditions**

Measurement view: Y Factor

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### **Group**

Calculate commands

### **Syntax**

CALCulate:NOISe:YFACTOR:MARKer<x>:DELTA:X?

### **Related Commands**

CALCulate:NOISe:YFACTOR:MARKer<x>:DELTA:Y?

### **Returns**

<NR2> = a floating point value without an exponent. This is the frequency difference value between MR and the selected marker.



## Examples

`CALC:NOIS:YFAC:MARK1:DELTA:X?` might return 100.0000, indicating the frequency difference between MR (reference marker) and the selected marker(M1) is 100 dB.

## **CALCulate:NOISe:YFACtor:MARKer<x>:DELTA:Y? (Query Only)**

Queries the power difference value between MR (reference marker) and the selected marker.

### Conditions

Measurement view: Y Factor

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid.

### Group

Calculate commands

### Syntax

```
CALCulate:NOISe:YFACtor:MARKer<x>:DELTA:Y?
```

### Related Commands

```
CALCulate:NOISe:YFACtor:MARKer<x>:DELTA:X?
```

### Returns

<NR2> = a floating point value without an exponent. This is the power difference value between MR and the selected marker.

## Examples

```
CALC:NOIS:YFAC:MARK1:DELTA:Y?
```

might return 18.7647705078, indicating the difference between MR (reference marker) and the selected marker (M1) is 18.76 dB.

## **CALCulate:NOISe:YFACtor:MARKer<x>:MAXimum (No Query Form)**

Places the selected marker at the maximum value on the peak of the trace.

### Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

### Group

Calculate commands

### Syntax

```
CALCulate:NOISe:YFACtor:MARKer<x>:MAXimum
```

## Related Commands

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:HIGHer?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:HIGHer? (Query Only)

This command places the selected marker at the next highest peak on the trace.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:HIGHer?
```

## Related Commands

CALCulate:NOISe:YFACTOR:MARKer<x>:MAXimum

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:RIGHT?

## Arguments

None

## Returns

None

## CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LEFT? (Query Only)

This command places the selected marker at the next peak on the trace to the left of the selected marker.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:LEFT?
```

## Related Commands

```
CALCulate:NOISe:YFActor:MARKer<x>:MAXimum
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:HIGHer?
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:LOWer?
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:RIGHT?
```

## Arguments

None

## Returns

None

## CALCulate:NOISe:YFActor:MARKer<x>:PEAK:LOWer? (Query Only)

Places the selected marker at the next lowest peak on the trace.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:LOWer?
```

## Related Commands

```
CALCulate:NOISe:YFActor:MARKer<x>:MAXimum
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:LEFT?
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:HIGHer?
```

```
CALCulate:NOISe:YFActor:MARKer<x>:PEAK:RIGHT?
```

## Arguments

None

## Returns

None

## CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:RIGHT? (Query Only)

Places the specified marker on a specified trace. The query form returns the trace on which the marker resides.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:RIGHT?
```

## Related Commands

CALCulate:NOISe:YFACTOR:MARKer<x>:MAXimum

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LEFT?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:LOWer?

CALCulate:NOISe:YFACTOR:MARKer<x>:PEAK:HIGHer?

## Arguments

None

## Returns

None

## CALCulate:NOISe:YFACTOR:MARKer<x>:TRACe

Sets or queries which trace the specified marker is attached to.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFACTOR:MARKer<x>:TRACe {TRACE1|TRACE2|TRACE3}
```

```
CALCulate:NOISe:YFACTOR:MARKer<x>:TRACe?
```

## Arguments

TRACE1 moves the specified marker to trace 1.

TRACE2 moves the specified marker to trace 2.

TRACE3 moves the specified marker to trace 3.

## Returns

TRACE1 means the specified marker is on trace 1.

TRACE2 means the specified marker is on trace 2.

TRACE3 means the specified marker is on trace 3.

## Examples

CALC:NOIS:YFAC:MARK1:TRAC? might return TRACE2, indicating that M1(marker 1) is on trace 2.

## CALCulate:NOISe:YFACTOR:MARKer<x>:X? (Query Only)

Queries the frequency of the specified marker.

## Conditions

Measurement view: Y Factor

The parameter <x> = 0 to 4

## Group

Calculate commands

## Syntax

```
CALCulate:NOISe:YFACTOR:MARKer<x>:X?
```

## Related Commands

CALCulate:NOISe:YFACTOR:MARKer<x>:Y?

## Returns

<NR3> = a floating point value with an exponent. This is the frequency of the specified marker.

## Examples

CALC:NOIS:YFAC:MARK1:X? might return 608.000000000E+6, indicating the frequency of M1 (marker 1) is 608 MHz.

## **CALCulate:NOISe:YFACTOR:MARKer<x>:Y? (Query Only)**

This command queries the vertical position (dB) of the specified marker.

### **Conditions**

Measurement view: Y Factor

The parameter <x> = 0 to 4

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:NOISe:YFACTOR:MARKer<x>:Y?
```

### **Related Commands**

```
CALCulate:NOISe:YFACTOR:MARKer<x>:X?
```

### **Returns**

<NR2> = a floating point value without an exponent. This is the dB of the specified marker.

### **Examples**

CALCULATE:NOISE:YFACTOR:MARKER1:Y? might return 18.7647705078, indicating the value of M1 (marker 1) is 18.76 dB.

## **CALCulate:OBWidth:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker frequency for the selected marker in the Occupied Bandwidth measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:OBWidth:MARKer<x>:DELTA:X?
```

### **Related Commands**

```
CALCulate:OBWidth:MARKer<x>:DELTA:Y?
```

### **Arguments**

None

## Returns

<NRf> Delta marker frequency for the selected marker.

## Examples

CALCULATE:OBWIDTH:MARKER1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

## CALCulate:OBWidth:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker in the Occupied Bandwidth measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>:DELTA:Y?

## Related Commands

CALCulate:OBWidth:MARKer<x>:DELTA:X?

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCULATE:OBWIDTH:MARKER1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## CALCulate:OBWidth:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

`CALCulate:OBWidth:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALCULATE:OBWIDTH:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Occupied Bandwidth trace.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

`CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer`

## Related Commands

`CALCulate:OBWidth:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALCULATE:OBWIDTH:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:OBWidth:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

`CALCulate:OBWidth:MARKer<x>:PEAK:LEFT`



---

## Related Commands

CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT

## Arguments

None

## Examples

CALCULATE:OBWIDTH:MARKER1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:OBWidth:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Occupied Bandwidth trace.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>:PEAK:LOWer

## Related Commands

CALCulate:OBWidth:MARKer<x>:PEAK:HIGHer

## Arguments

None

## Examples

CALCULATE:OBWIDTH:MARKER1:PEAK:LOWER moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>:PEAK:RIGHT

## Related Commands

CALCulate:OBWidth:MARKer<x>:PEAK:LEFT

## Arguments

None

## Examples

CALCULATE:OBWIDTH:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:OBWidth:MARKer<x>[:SET]:CENTER (No Query Form)

Sets the center frequency to the value at the marker position in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>[:SET]:CENTER

## Arguments

None

## Examples

CALCULATE:OBWIDTH:MARKER1:SET:CENTER sets the center frequency to the value at the Marker 1 position.

## CALCulate:OBWidth:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>:X <value>

CALCulate:OBWidth:MARKer<x>:X?

## Related Commands

CALCulate:OBWidth:MARKer<x>:Y?

## Arguments

<value>::=<NRf> specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

CALCULATE:OBWIDTH:MARKER1:X 800MHz places Marker 1 (M1) at 800 MHz on the trace.

## CALCulate:OBWidth:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Calculate commands

## Syntax

CALCulate:OBWidth:MARKer<x>:Y?

## Related Commands

CALCulate:OBWidth:MARKer<x>:X

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

CALCULATE:OBWIDTH:MARKER1:Y? might return -34.28 indicating Marker 1(M1) is at -34.28 dBm.

## CALCulate:OFDM:CONSt:MARKer<x>:FREQuency

Sets or queries the frequency position of the selected marker in the OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

```
CALCulate:OFDM:CONStE:MARKer<x>:FREQuency <value>
```

```
CALCulate:OFDM:CONStE:MARKer<x>:FREQuency?
```

## Arguments

<value> : :=<NRf> specifies the frequency position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Returns

<NRf> The value type depends on where the marker is placed.

## Examples

CALCulate:OFDM:CONStE:MARKer1:FREQuency? might return 800 MHz, indicating that Marker 1 M1 is positioned at the 800 MHz position in the OFDM Constellation view.

## CALCulate:OFDM:CONStE:MARKer<x>:MAGNitude? (Query Only)

Queries the marker magnitude of the selected marker in the OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

```
CALCulate:OFDM:CONStE:MARKer<x>:MAGNitude?
```

## Arguments

None

## Returns

<NRf> The magnitude for the selected marker in dB.

## Examples

CALCulate:OFDM:CONStE:MARKer1:MAGNitude might return 35 indicating readout of Marker 1 (M1) is 35 dB.

## CALCulate:OFDM:CONStE:MARKer<x>:PHASe? (Query Only)

Queries the phase readout of the selected marker in the OFDM Constellation measurement view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

```
CALCulate:OFDM:CONStE:MARKer<x>:PHASe?
```

## Arguments

None

## Returns

<NRf> The phase readout for the selected marker.

## Examples

`CALCulate:OFDM:CONStE:MARKer1:PHASe?` might return 35.74, indicating the phase readout of Marker 1 (M1) is 35.74°.

## CALCulate:OFDM:CONStE:MARKer<x>:TIME

Sets or queries the X position of the selected marker in the OFDM Constellation measurement view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

```
CALCulate:OFDM:CONStE:MARKer<x>:TIME <value>
```

```
CALCulate:OFDM:CONStE:MARKer<x>:TIME?
```

## Arguments

<value::=NRf> specifies the marker time.

## Examples

`CALCulate:OFDM:CONStE:MARKer1:TIME -234.5us` places Marker 1 (M1) at the -234.5  $\mu$ s on the trace.

## CALCulate:OFDM:CONStE:MARKer<x>:TYPE? (Query Only)

Queries the data type of the selected marker in OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:CONStE:MARKer<x>:TYPE?`

## Arguments

None

## Returns

1.000 if the marker type is a data marker.

0.000 if the marker type is a pilot marker.

## Examples

`CALCulate:OFDM:CONStE:MARKer1:TYPE?` might return 1.000 indicating Marker 1 (M1) is a data marker.

## **CALCulate:OFDM:CONStE:MARKer<x>:VALue? (Query Only)**

Queries the value of the specified marker in OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:CONStE:MARKer<x>:VALue?`

## Arguments

None

## Returns

<NRf> The value readout for the selected marker.

## Examples

`CALCulate:OFDM:CONStE:MARKer2:VALue?` might return 2.00, indicating the value readout of Marker 2 (M2) is 2.

## **CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker frequency for the selected marker in the OFDM Channel response view.

### **Conditions**

Measurement views: OFDM

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:X?
```

### **Arguments**

None

### **Returns**

<NRf> Delta Marker frequency for the selected marker.

### **Examples**

```
CALCulate:OFDM:CRESPonse:MARKer2:DELTA:X
```

might return 1.28E+6 indicating that the delta marker frequency for Marker 2 (M2) is 1.28 MHz.

## **CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker frequency for the selected marker in the OFDM Channel response view.

### **Conditions**

Measurement views: OFDM

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:OFDM:CRESPonse:MARKer<x>:DELTA:Y?
```

### **Arguments**

None

### **Returns**

<NRf> Delta Marker frequency for the selected marker.

## Examples

`CALCulate:OFDM:CRESpOse:MARKer2:DELTA:Y` might return `1.28E+6` indicating that the delta marker frequency for Marker 2 (M2) is 1.28 MHz.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak in the OFDM Channel response view.

### Conditions

Measurement views: OFDM

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:CRESpOse:MARKer<x>:MAXimum`

### Arguments

None

## Examples

`CALCulate:OFDM:CRESpOse:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak in the OFDM Channel response view.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude in the OFDM Channel response view.

### Conditions

Measurement views: OFDM

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:HIGHer`

### Arguments

None



## Examples

`CALCulate:OFDM:CRESpOse:MARKer1:PEAK:HIGHer` moves Marker 1(M1) to the next peak higher in amplitude in the OFDM Channel response view.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left in the OFDM Channel response view.

## Conditions

Measurement views: OFDM

## Measurement views: OFDM

Calculate commands

## Syntax

`CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:LEF`

## Arguments

None

## Examples

`CALCulate:OFDM:CRESpOse:MARKer1:PEAK:LEFT` moves Marker1 (M1) to the next peak to the left in the OFDM Channel response view.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude in the OFDM Channel response view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALCulate:OFDM:CRESpOse:MARKer2:PEAK:LOWer` moves Marker 2 (M2) to the next peak lower in amplitude in the OFDM Channel response view.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right in the OFDM Channel response view.

### **Conditions**

Measurement views: OFDM

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:OFDM:CRESpOse:MARKer<x>:PEAK:RIGHT
```

### **Arguments**

None

### **Examples**

`CALCulate:OFDM:CRESpOse:MARKer1:PEAK:RIGHT` moves Marker1 (M1) to the next peak to the right in the OFDM Channel response view.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker in the OFDM Channel response view.

### **Conditions**

Measurement views: OFDM

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:OFDM:CRESpOse:MARKer<x>:X <value>
```

```
CALCulate:OFDM:CRESpOse:MARKer<x>:X?
```

### **Arguments**

`<value> ::= <NRf>` specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

### **Examples**

`CALCulate:OFDM:CRESpOse:MARKer1:X 800MHZ` places Marker 1 (M1) at

800 MHz on the trace.

## **CALCulate:OFDM:CRESpOse:MARKer<x>:Y? (Query Only)**

Returns the amplitude of the selected marker in the OFDM Channel response view.

### **Conditions**

Measurement views: OFDM

### **Group**

Calculate commands

### **Syntax**

`CALCulate:OFDM:CRESpOse:MARKer<x>:Y?`

### **Arguments**

None

### **Returns**

<NRf> Amplitude of the selected marker in dB.

### **Examples**

`CALCulate:OFDM:CRESpOse:MARKer1:Y` might return `33.393856E-3` indicating the amplitude of Marker 1 (M1) is 0.033 dB.

## **CALCulate:OFDM:FLATness:MARKer<x>:MAXimum (No Query Form)**

Positions the specified marker at the maximum point on the trace in the OFDM Spectral Flatness display.

### **Conditions**

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### **Group**

Calculate commands

### **Syntax**

`CALCulate:OFDM:FLATness:MARKer<x>:MAXimum`

### **Arguments**

None

## Examples

`CALC:OFDM:FLAT:MARK2:MAX` positions the Marker 2 (M2) at the maximum point on the trace.

## **CALCulate:OFDM:FLATness:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the specified marker to the next peak higher in amplitude on the trace on the OFDM Spectral Flatness display.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:PEAK:HIGHer`

### Arguments

None

## Examples

`CALC:OFDM:FLAT:MARK2:PEAK:HIGHer` moves Marker 2 (M2) to the next peak higher in amplitude on the trace.

## **CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the specified marker to the next peak to the left on the trace in the OFDM Spectral Flatness display.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LEFT`

### Arguments

None

## Examples

`CALC:OFDM:FLAT:MARK2:PEAK:LEFT` moves Marker 2 (M2) to the next peak to the left on the trace .

## **CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the marker to the next peak lower in amplitude on the trace on the OFDM Spectral Flatness display.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:PEAK:LOWer`

### Arguments

None

## Examples

`CALC:OFDM:FLAT:MARK2:PEAK:LOW` moves Marker 2 (M2) to the next peak lower on the trace.

## **CALCulate:OFDM:FLATness:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the marker to the next peak to the right on the trace on the OFDM Spectral Flatness display.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:PEAK:RIGHT`

### Arguments

None

## Examples

`CALC:OFDM:FLAT:MARK2:PEAK:RIGHT` moves Marker 2 (M2) to the next peak to the next peak on the right on the trace.

## **CALCulate:OFDM:FLATness:MARKer<x>:X**

Sets or queries the horizontal value at the selected marker position on the points trace in the OFDM Spectral Flatness display. The units are in Frequency or Subcarrier.

To set the units, use the command `[SENSe]:OFDM:UNIT:FREQuency`.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:X <NRf>`

`CALCulate:OFDM:FLATness:MARKer<x>:X?`

## Arguments

Floating point number that specifies the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

`CALC:OFDM:FLAT:MARK3:X 22` places Marker 3 at 22 Subcarrier on the trace (assuming the units have been set to Subcarrier).

## **CALCulate:OFDM:FLATness:MARKer<x>:Y? (Query Only)**

Returns the value of the amplitude (vertical position) at the selected marker position on the OFDM Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:FLATness:MARKer<x>:Y?`

## Returns

Floating point number that indicates the amplitude at the selected marker position in dB.

## Examples

`CALC:OFDM:FLAT:MARK1:Y?` might return `33.393856E-3` indicating the amplitude at Marker 1 is 0.033 dB.

## CALCulate:OFDM:TABLE:MARKer<x>:FREQuency

Sets or queries the frequency position of the selected marker in the OFDM Symbol table view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:TABLE:MARKer<x>:FREQuency <value>`

`CALCulate:OFDM:TABLE:MARKer<x>:FREQuency?`

## Arguments

`<value>::=<NRf>` specifies the marker frequency.

## Examples

`CALCulate:OFDM:TABLE:MARKer1:FREQuency 800MHZ` places Marker 1(M1) at 800 MHz in the symbol table.

## CALCulate:OFDM:TABLE:MARKer<x>:TIME

Sets or queries the marker time in the OFDM Symbol table view.

## Conditions

Measurement views: OFDM

## Group

Calculate commands

## Syntax

`CALCulate:OFDM:TABLE:MARKer<x>:TIME <value>`

`CALCulate:OFDM:TABLE:MARKer<x>:TIME?`

## Arguments

`<value>::=<NRf>` specifies the marker time.

## Examples

`CALCulate:OFDM:TABLE:MARKer1:TIME -234.5us` places Marker 1 (M1) at  $-234.5 \mu\text{s}$  in the symbol table.

## CALCulate:OFDM:TABLE:MARKer<x>:VALue? (Query Only)

Queries the value readout of the selected marker in the OFDM Symbol table view.

### Conditions

Measurement views: OFDM

### Group

Calculate commands

### Syntax

`CALCulate:OFDM:TABLE:MARKer<x>:VALue?`

### Arguments

None

### Returns

<NRf> The value readout of the selected marker.

## Examples

`CALCulate:OFDM:TABLE:MARKer2:VALue?` might return 2.00 indicating the value readout of Marker 2 (M2) is 2.

## CALCulate:P25:CONStE:MARKer<x>:DELTA:X[:TIME]? (Query Only)

Returns the delta marker time for the selected marker on the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation

The parameter <x>=1 to 4; MARKer0 (reference marker) is invalid.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Group

Calculate commands

### Syntax

`CALCulate:P25:CONStE:MARKer<x>:DELTA:X[:TIME]?`

### Returns

<NR2> = a floating point value without an exponent. This is the time for the selected Delta marker.





**Note:** Use the `[SENSe]:P25:TIME:UNITs` command to select the time unit: symbols (default) or seconds.

## Examples

`CALCulate:P25:CONStE:MARKer1:DELTA:X:TIME?` might return 62.75, indicating that the Delta marker time is 62.75 symbols.

## CALCulate:P25:CONStE:MARKer<x>:FDEVIation? (Query Only)

Queries the frequency deviation of the selected marker in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation

The parameter `<x>` =0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Group

Calculate commands

### Syntax

`CALCulate:P25:CONStE:MARKer<x>:FDEVIation?`

### Returns

`<NR2>` = a floating point value without an exponent. This is the frequency deviation readout for the selected marker.

## Examples

`CALC:P25:CONS:MARr1:FDEV?` might return 102.82380000E+3, indicating the frequency deviation readout of Marker 1 (M1) is 102.824 kHz.

## CALCulate:P25:CONStE:MARKer<x>:MAGNitude? (Query Only)

Queries the frequency deviation (for Freq Dev trace types) or magnitude (for IQ trace types) of the selected marker in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation

### Group

Calculate commands

### Syntax

`CALCulate:P25:CONStE:MARKer<x>:MAGNitude?`

## Returns

<NRf> is the frequency deviation or magnitude of the selected marker.

## Examples

`CALC:P25:CONS:MARK1:MAGN?` might return `1.6257750459E+3` if the trace type is Freq Dev, indicating that Marker 1 (M1) has a frequency deviation of 1.626 kHz.

`CALC:P25:CONS:MARK1:MAGN?` might return `1.0048475788` if the trace type is IQ, indicating that Marker 1 (M1) has a magnitude of 1.005.

## CALCulate:P25:CONStE:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Group

Calculate commands

### Syntax

`CALCulate:P25:CONStE:MARKer<x>:MAXimum`

### Related Commands

`CALCulate:P25:CONStE:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:CONStE:MARKer<x>:PEAK:RIGHT`

### Examples

`CALC:P25:CONS:MARK1:MAX` positions Marker 1 (M1) at the symbol in the center of the time record.

## CALCulate:P25:CONStE:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation

### Group

Calculate commands

### Syntax

`CALCulate:P25:CONStE:MARKer<x>:PEAK:LEFT`

## Related Commands

CALCulate:P25:CONStE:MARKer<x>:PEAK:RIGHT

CALCulate:P25:CONStE:MARKer<x>:MAXimum

## Examples

CALC:P25:CONS:MARK1:PEAK:LEFT moves Marker 1 (M1) in the time domain to the next lower symbol number.

## CALCulate:P25:CONStE:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

The specified marker must be activated using the CALCulate:MARKer:ADDcommand.

## Group

Calculate commands

## Syntax

CALCulate:P25:CONStE:MARKer<x>:PEAK:RIGHT

## Related Commands

CALCulate:P25:CONStE:MARKer<x>:MAXimum

CALCulate:P25:CONStE:MARKer<x>:PEAK:LEFT

## Examples

CALC:P25:CONS:MARK1:PEAK:RIGHT moves theMarker 1 (M1) in the time domain to the next higher symbol number to the right.

## CALCulate:P25:CONStE:MARKer<x>:PHASe? (Query Only)

Queries the phase of the selected marker in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

IQ trace type required.

## Group

Calculate commands

## Syntax

CALCulate:P25:CONStE:MARKer<x>:PHASe?

## Returns

<NR2> = a floating point value without an exponent. This is the phase value for the selected marker.

## Examples

`CALC:P25:CONS:MARK1:PHAS?` might return 41.4721108251, indicating the phase of Marker 1 (M1) is 41.5°.

## CALCulate:P25:CONSte:MARKer<x>:SYMBol? (Query Only)

Queries the readout symbol of the selected marker in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

The parameter <x>= 1 to 4; MARKer0 (reference marker) is invalid.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

`CALCulate:P25:CONSte:MARKer<x>:SYMBol?`

## Returns

<NR2> = a floating point value is the readout symbol for the selected marker.

## Examples

`CALCulate:P25:CONSte:MARKer1:SYMBol?` might return 62.0000000000, indicating the readout symbol of Marker 1 (M1) is 62.

## CALCulate:P25:CONSte:MARKer<x>:Trace

Places the selected marker on the Demodulated I and Q versus Time trace. The query returns the name of the trace on which the marker resides.

## Conditions

Measurement view: P25 Constellation

The parameter <x>= 0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:CONStE:MARKer<x>:Trace {Trace1 | Trace2}
```

```
CALCulate:P25:CONStE:MARKer<x>:Trace?
```

## Arguments

Trace 1: sets the I trace marker on.

Trace 2: sets the Q trace marker on.

## Returns

Trace 1 = I trace

Trace 2 = Q trace

## Examples

`CALC:P25:CONS:MARK1:TRAC` Trace1 places Marker 1 (M1) on the I trace.

## CALCulate:P25:CONStE:MARKer<x>:VALue? (Query Only)

Queries the readout value of the specified marker in P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

## Group

Calculate commands

## Syntax

```
CALCulate:P25:CONStE:MARKer<x>:VALue?
```

## Returns

<NRf> is the readout value for the selected marker.

## Examples

`CALC:P25:CONS:MARK2:VAL?` might return 2.00, indicating the readout value of Marker 2 (M2) is 2.

## CALCulate:P25:CONStE:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

## Group

Calculate commands

## Syntax

```
CALCulate:P25:CONStE:MARKer<x>:X <value>
```

```
CALCulate:P25:CONStE:MARKer<x>:X?
```

## Arguments

<value> : :=<NRf> specifies the time position of the marker. Range is (analysis offset) to [(analysis offset) + (analysis length)]

## Returns

Offset set value(-500Hz to +500Hz).

## Examples

`CALCulate:P25:PVTime:MARKer1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace. When used as a query, the return will show the horizontal position of the trace. For example, 800 MHz.

## CALCulate:P25:EDIagram:MARKer<x>:DELTA:X[:TIME]? (Query Only)

Returns the delta marker time for the selected marker on the P25 Eye Diagram trace.

## Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4. MARKer0 (reference marker) is invalid.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:DELTA:X[:TIME]?
```

## Related Commands

```
CALCulate:P25:EDIagram:MARKer<x>:DELTA:Y?
```

## Returns

<NRf> is the Delta marker time for the selected marker. Use the `[SENSe]:P25:TIME:UNITs` command to select the time unit: symbols (default) or seconds.

## Examples

`CALCulate:P25:EDIagram:MARKer1:DELTA:X:TIME?` might return 62.750000000, indicating that the delta marker time is 62.75 symbols.

## CALCulate:P25:EDIagram:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the P25 Eye Diagram trace.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=1 to 4. MARKer0 (reference marker) is invalid.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:DELTA:Y?
```

### Related Commands

```
CALCulate:P25:EDIagram:MARKer<x>:DELTA:X[:TIME]?
```

### Returns

<NRf> is the Delta marker amplitude for the selected marker.

### Examples

CALCulate:P25:EDIagram:MARKer1:DELTA:Y? might return -1.04300000, indicating that the delta marker amplitude is -1.043.

## CALCulate:P25:EDIagram:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the P25 Eye Diagram trace.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:MAXimum
```

### Related Commands

```
CALCulate:P25:EDIagram:MARKer<x>:PEAK:HIGHer
```

CALCulate:P25:EDiagram:MARKer<x>:PEAK:LEFT  
CALCulate:P25:EDiagram:MARKer<x>:PEAK:LOWer  
CALCulate:P25:EDiagram:MARKer<x>:PEAK:RIGHT

### Arguments

None

### Returns

None

### Examples

CALCulate:P25:EDiagram:MARKer1:MAXimum moves Marker 1 (M1) to the highest peak on the trace.

## CALCulate:P25:EDiagram:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the P25 Eye Diagram trace.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

CALCulate:P25:EDiagram:MARKer<x>:PEAK:HIGHer

### Related Commands

CALCulate:P25:EDiagram:MARKer<x>:MAXimum  
CALCulate:P25:EDiagram:MARKer<x>:PEAK:LEFT  
CALCulate:P25:EDiagram:MARKer<x>:PEAK:LOWer  
CALCulate:P25:EDiagram:MARKer<x>:PEAK:RIGHT

### Arguments

None

### Returns

None



## Examples

`CALCulate:P25:EDIagram:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:P25:EDIagram:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the P25 Eye Diagram trace.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Group

Calculate commands

### Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:PEAK:LEFT
```

### Related Commands

`CALCulate:P25:EDIagram:MARKer<x>:MAXimum`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:HIGHer`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:LOWer`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:RIGHT`

### Arguments

None

### Returns

None

## Examples

`CALCulate:P25:EDIagram:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:P25:EDIagram:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the P25 Eye Diagram trace.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:PEAK:LOWer
```

## Related Commands

`CALCulate:P25:EDIagram:MARKer<x>:MAXimum`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:HIGHer`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Returns

None

## Examples

`CALCulate:P25:EDIagram:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:P25:EDIagram:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the P25 Eye Diagram trace.

## Conditions

Measurement view: P25 Eye Diagram

The parameter `<x>`=0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:PEAK:RIGHT
```

## Related Commands

`CALCulate:P25:EDIagram:MARKer<x>:MAXimum`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:EDIagram:MARKer<x>:PEAK:HIGHer`

CALCulate:P25:EDIagram:MARKer<x>:PEAK:LOWer

### Arguments

None

### Returns

None

### Examples

CALCulate:P25:EDIagram:MARKer1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:P25:EDIagram:MARKer<x>:TRACe

Sets the selected marker on the I or Q trace, or queries the trace that the specified marker is on, in the P25 Eye Diagram display.

### Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:TRACe { TRACe1 | TRACe2 }
```

```
CALCulate:P25:EDIagram:MARKer<x>:TRACe?
```

### Arguments

TRACe1 places the specified marker on the I trace.

TRACe2 places the specified marker on the Q trace.

### Returns

TRACe1 means the specified marker is on the I trace.

TRACe2 means the specified marker is on the Q trace.

### Examples

CALCulate:P25:EDIagram:MARKer1:TRACe TRACe1 places Marker 1 (M1) on the I trace in the P25 Eye Diagram display.

## CALCulate:P25:EDIagram:MARKer<x>:X

Sets or queries the horizontal position (time) of the selected marker in the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:X <NRf>
```

```
CALCulate:P25:EDIagram:MARKer<x>:X?
```

## Related Commands

```
CALCulate:P25:EDIagram:MARKer<x>:Y
```

## Arguments

<NRf> specifies the horizontal position (time) of the marker.



**Note:** Use the `[SENSe]:P25:TIME:UNITs` command to select the time unit: Symbols (default) or seconds.

## Returns

<NRf> is the horizontal position of the selected marker.

## Examples

```
CALC:P25:EDI:MARK1:X 38.5
```

 places Marker 1 (M1) at 38.5 symbols on the trace.

## CALCulate:P25:EDIagram:MARKer<x>:Y

Sets or queries the vertical position of the selected marker in the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

The parameter <x>=0 to 4.

The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:EDIagram:MARKer<x>:Y <value>
```

```
CALCulate:P25:EDIagram:MARKer<x>:Y?
```

## Related Commands

CALCulate:P25:EDiagram:MARKer<x>:X

## Arguments

<value>:: =<NRf> specifies the vertical position of the marker.

## Returns

<NRf> is the vertical position of the selected marker.

## Examples

CALC:P25:EDI:MARK1:Y? might return 571.8E-3, indicating Marker 1 (M1) is at 0.5718.

## CALCulate:P25:PVTime:MARKer<x>:DELTA:X[:TIME]? (Query Only)

Returns the delta marker time (seconds) for the selected marker on the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

The parameter <x>= 1 to 4; MARKer0 (reference marker) is invalid.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Group

Calculate commands

## Syntax

CALCulate:P25:PVTime:MARKer<x>:DELTA:X[:TIME]?

## Related Commands

CALCulate:P25:PVTime:MARKer<x>:DELTA:Y[:TIME]? SENSE:P25:TIME:UNITs

## Returns

<NRf> is the Delta marker time for the selected marker.



**Note:** Use the SENSE:P25:TIME:UNITs command to select the time unit symbols (default) or seconds.

## Examples

CALC:P25:PVT:MARK1:DELTA:X? might return -3.99372800000E-3, indicating that the delta marker 1 (M1) time is -3.937 ms.

## CALCulate:P25:PVTime:MARKer<x>:DELTA:Y[:TIME]? (Query Only)

Returns the Delta marker amplitude (dB) for the selected marker on the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

The parameter <x>= 1 to 4; MARKer0 (reference marker) is invalid.

The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Group

Calculate commands

## Syntax

```
CALCulate:P25:PVTime:MARKer<x>:DELTA:Y[:TIME]?
```

## Related Commands

```
CALCulate:P25:PVTime:MARKer<x>:DELTA:X[:TIME]?
```

## Returns

<NR3> is a floating point value with an exponent. This is the Delta marker amplitude.

## Examples

CALC:P25:PVT:MARK1:DELTA:Y? might return 552.6123046875E-3, indicating that the Delta marker 1 (M1) amplitude is 0.55 dB.

## CALCulate:P25:PVTime:MARKer<x>:Maximum (No Query Form)

Moves the selected marker to the highest peak on the trace in the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:P25:PVTime:MARKer<x>:Maximum
```

## Related Commands

```
CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer
```

```
CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer
```

```
CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT
```

```
CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Returns

None

## Examples

`CALC:P25:PVT:MARK1:PEAK:MAX` moves Marker 1 (M1) to the highest peak on the trace.

## CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the trace in the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer`

## Related Commands

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer`

`CALCulate:P25:PVTime:MARKer<x>:Maximum`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Returns

None

## Examples

`CALC:P25:PVT:MARK1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the trace in the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT`

## Related Commands

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer`

`CALCulate:P25:PVTime:MARKer<x>:Maximum`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Returns

None

## Examples

`CALC:P25:PVT:MARK1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the trace in the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer`

## Related Commands

`CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer`

`CALCulate:P25:PVTime:MARKer<x>:Maximum`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT`

## Arguments

None



## Returns

None

## Examples

`CALC:P25:PVT:MARK1:PEAK:LOW` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the trace in the Power vs. Time display.

## Conditions

Measurement view: Power vs. Time

## Group

Calculate commands

## Syntax

`CALCulate:P25:PVTime:MARKer<x>:PEAK:RIGHT`

## Related Commands

`CALCulate:P25:PVTime:MARKer<x>:PEAK:HIGHer`

`CALCulate:P25:PVTime:MARKer<x>:Maximum`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LEFT`

`CALCulate:P25:PVTime:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Returns

None

## Examples

`CALC:P25:PVT:MARK1:PEAK:RIGH` moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:P25:PVTime:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:P25:PVTime:MARKer<x>:X <value>
```

```
CALCulate:P25:PVTime:MARKer<x>:X?
```

## Related Commands

```
CALCulate:P25:PVTime:MARKer<x>:Y
```

## Arguments

<value> ::= <NRf> specifies the horizontal position of the marker. Range is Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Returns

<value> ::= <NRf> specifies the horizontal position of the marker.

## Examples

`CALC:P25:PVT:MARK1:X 800MHz` places Marker 1 (M1) at 800 MHz on the trace. When used as a query, the return will show the horizontal position of the trace. For example, 800 MHz.

## CALCulate:P25:PVTime:MARKer<x>:Y

Sets or queries the vertical position of the selected marker in the P25 Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Calculate commands

## Syntax

```
CALCulate:P25:PVTime:MARKer<x>:Y <value>
```

```
CALCulate:P25:PVTime:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:P25:PVTime:MARKer<x>:X
```

## Arguments

<value> ::= <NRf> specifies the vertical position of the marker

## Returns

<NR2> = floating point value without an exponent.

## Examples

`CALC:P25:PVT:MARK1:Y?` might return `-67.1001052856`, indicating that the horizontal position of Marker 1 (M1) is `-67.10 dBm`.

`CALC:P25:PVT:MARK1:Y -75.66dBm` places Marker 1 (M1) at `-75.66` on the trace.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:X? (Query only)**

Returns the delta marker time for the selected marker in the Phase error versus Time measurement.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

`MEASview` keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: `GPDemod Phase error versus Time`).

## Group

Calculate commands

## Syntax

`CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:X?`

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

`<NRf>` Delta marker time for the selected marker in percent (%).

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:DELTA:X?` might return `3.84`, indicating that the delta marker duration is `3.84` for measurement view 2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)**

Returns the delta marker phase for the selected marker in the Phase error versus Time measurement.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:DELTA:X?](#)

## Arguments

None

## Returns

<NRf> Delta marker phase for the selected marker in percent (%).

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:DELTA:Y?` might return 3.84, indicating that the delta marker amplitude is 3.84% for measurement view2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the trace in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace for measurement view2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the Phase error versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer](#)

## Arguments

None

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace for measurement view 2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the trace in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace for measurement view 2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the Phase error versus Time trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer](#)

## Arguments

None

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace for measurement view 2.

## **CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the trace in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Calculate commands

## Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:PERRor\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace for measurement view 2.

## CALCulate:PERRor[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Phase error versus Time measurement.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is not valid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:X?
```

### Arguments

<value> : :=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)]. The unit can be changed by [SENSe]:DDEMod:TIME:UNITs command.

### Examples

```
CALCulate:PERRor:MEASview2:MARKer1:X 1.5 µs places Marker 1 (M1) at 1.5 µs on the trace for measurement view 2.
```

## CALCulate:PERRor[:MEASview<y>]:MARKer<x>:Y? (Query only)

Queries the marker amplitude of the selected marker in the Phase error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### Group

Calculate commands

### Syntax

```
CALCulate:PERRor[:MEASview<y>]:MARKer<x>:Y?
```



## Related Commands

`CALCulate:PERRor[:MEASview<y>]:MARKer<x>:X`

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in percent (%).

## Examples

`CALCulate:PERRor:MEASview2:MARKer1:Y?` might return 5.34, indicating Marker 1 (M1) is at 5.34% for measurement view2.

## CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker time for the selected marker in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:X?`

## Related Commands

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

`CALCulate:PHVTime:MEASview1:MARKer1:DELTA:X?` might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker phase for the selected marker in the Phase versus Time measurement.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

### **Related Commands**

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:DELTA:X?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Returns**

<NRf> Delta marker phase for the selected marker in degrees.

### **Examples**

CALCULATE:PHVTIME:MEASview1:MARKER1:DELTA:Y? might return 162.38, indicating that the delta marker phase is 162.38 °.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the trace in the Phase versus Time measurement.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE:PHVTIME:MEASview1:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Phase versus Time trace.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

## Related Commands

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE:PHVTIME:MEASview1:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the trace in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT`

## Related Commands

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE:PHVTIME:MEASview1:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the Phase versus Time trace.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Related Commands

`CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE:PHVTIME:MEASview1:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the trace in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

CALCULATE:PHVTIME:MEASview1:MARKER1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Calculate commands

## Syntax

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:X?
```

## Related Commands

```
CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

<value>::=<NRf> specifies the horizontal position of the marker. Range: (analysis offset) to [(analysis offset) + (analysis length)].

## Examples

CALCULATE:PHVTIME:MEASview1:MARKER1:X 1.5US places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

## CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the Phase versus Time measurement.

### Conditions

Measurement views: Phase versus Time

### Group

Calculate commands

### Syntax

CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:Y?

### Related Commands

CALCulate:PHVTime[:MEASview<y>]:MARKer<x>:X

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

### Returns

<NRf> Marker amplitude of the selected marker in degrees.

## Examples

CALCULATE:PHVTIME:MEASview1:MARKER1:Y? might return -18.435, indicating Marker 1(M1) is at -18.435  $^{\circ}$ .

## CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:X

Sets or queries the horizontal position of the indicator on the trace in the Cumulative Histogram display. Where <x> parameters: 1 to 4

### Conditions

Measurement views: Pulse Cumulative Histogram with specified Measurement view number

### Group

Calculate commands

## Syntax

```
CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:X <value>
```

```
CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:X?
```

## Arguments

<value>::=<NRf> specifies the horizontal position of the marker.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Returns

<NRf>

## Examples

CALCulate:PULSe:CUMulative:HISTogram:MEASview1:INDicator:X 9.5409090173E-6 places the indicator at the nearest bins mid value at 9.540E-6 on the trace.

## CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:Y (Query Only)

Queries the vertical position value of the indicator on the trace in the Cumulative Histogram display.

## Conditions

Measurement views: Pulse Cumulative Histogram with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:CUMulative:HISTogram[:MEASview<y>]:INDicator:Y?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<value>

Where <value>::<NRf> is the vertical position value of the marker.

## Examples

CALCulate:PULSe:CUMulative:HISTogram:MEASview1:INDicator:Y might return 1.000000 for the vertical position of the marker in cumulative histogram Measurement View 1.

## **CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:X (Query Only)**

Returns the delta marker time for the selected marker in the Frequency vs Time display of Pulse-Ogram.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:X?
```

### **Arguments**

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Returns**

<NRf> Delta marker time for the selected marker.

### **Examples**

CALCulate:PULSe:OGRAM:MEASview1:MARKer1:FVTime:DELTA:X? might return 38.0E-6, indicating that the delta marker time for M1 is 38.0  $\mu$ s.

## **CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:Y (Query Only)**

Returns the delta marker amplitude for the selected marker in the Frequency vs Time display of Pulse-Ogram.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:DELTA:Y?
```

### **Arguments**

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.



<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> Delta marker Amplitude for the selected marker

## Examples

CALCulate:PULSe:OGRAM:MEASview1:MARKer1:FVTime:DELTA:Y? might return 45.82, indicating that the delta marker amplitude for Marker 1 (M1) is 45.82.

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in Frequency vs Time display of Pulse-Ogram.

## Conditions

Measurement views: Pulse-Ogram

## Group

Calculate commands

## Syntax

CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:MAXimum

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

CALCulate:PULSe:OGRAM:MEASview1:MARKer1:FVTime:MAXimum moves Marker 1(M1) to the highest peak on the trace.

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:X

Sets or Queries the horizontal position of the selected marker in Frequency vs Time display of Pulse-Ogram.

## Conditions

Measurement views: Pulse-Ogram

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:X <value>
```

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:X?
```

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>=<NRf> specifies the horizontal position of the marker.

## Returns

<NRf>

## Examples

```
CALCulate:PULSe:OGRAM:MEASview1:MARKer1:FVTime:X 2E-6 places Marker 1 (M1) at 2 μs on the Frequency vs Time trace.
```

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:Y (Query Only)

Returns the marker amplitude of the selected marker in Frequency vs Time display of Pulse-Ogram.

## Conditions

Measurement views: Pulse-Ogram

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:FVTime:Y?
```

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

`CALCulate:PULSe:OGRAM:MEASview1:MARKer1:FVTime:Y?` might return `-34.28`, indicating Marker 1 (M1) is at the amplitude of `-34.28` dBm.

## **CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:DELTA:X (Query Only)**

Returns the delta marker time for the selected marker in the Time vs Time display of Pulse-Ogram.

### Conditions

Measurement views: Pulse-Ogram

### Group

Calculate commands

### Syntax

`CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:DELTA:X ?`

### Arguments

`<x>` = 1, 2, 3, 4 is the marker. `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

`<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument. where `MEASview` keyword is optional by default `y` is 1.

### Returns

`<NRf>` Delta marker time for the selected marker

## Examples

`CALCulate:PULSe:OGRAM:MEASview1:MARKer1:TVTime:DELTA:X?` might return `-197.57E-6`, indicating that the delta marker time for Marker 1 (M1) is `197.57`  $\mu$ s.

## **CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:DELTA:Y (Query Only)**

Queries the delta marker amplitude for the selected marker in the Time vs Time display of Pulse-Ogram.

### Conditions

Measurement views: Pulse-Ogram

### Group

Calculate commands

## Syntax

`CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:DELTA:Y ?`

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> Delta marker amplitude for the selected marker

## Examples

`CALCulate:PULSe:OGRAM:MEASview1:MARKer1:TVTime:DELTA:Y?` might return `-813.10E-3`, indicating that the delta marker amplitude for Marker 1 (M1) is `-813.10E-3`.

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:MAXimum (No Query Form)

Moves the selected marker to the highest peak on the trace in Time vs Time display of Pulse-Ogram.

## Conditions

Measurement views: Pulse-Ogram

## Group

Calculate commands

## Syntax

`CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:MAXimum`

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

None

## Examples

`CALCulate:PULSe:OGRAM:MEASview1:MARKer1:TVTime:MAXimum` moves Marker 1(M1) to the highest peak on the trace of PulseOgram Measurement View 1.

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:X

Sets or Queries the horizontal position of the selected marker in Time vs Time display of Pulse-Ogram.

### Conditions

Measurement views: Pulse-Ogram

### Group

Calculate commands

### Syntax

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:X <value>
```

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:X ?
```

### Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>=<NRf> specifies the horizontal position of the marker,

### Returns

<NRf>

### Examples

CALCulate:PULSe:OGRAM:MEASview1:MARKer1:TVTime:X 2E-6 places Marker 1 (M1) at 2  $\mu$ s on the Time vs Time trace of PulseOgram Measurement View 1.

## CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:Y (Query Only)

Returns the marker amplitude of the selected marker in Time vs Time display of Pulse-Ogram.

### Conditions

Measurement views: Pulse-Ogram

### Group

Calculate commands

### Syntax

```
CALCulate:PULSe:OGRAM[:MEASview<y>]:MARKer<x>:TVTime:Y?
```

## Arguments

<x> = 1, 2, 3, 4 is the marker. MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf>Marker amplitude of the selected marker

## Examples

CALCulate:PULSe:OGRAM:MEASview1:MARKer1:TVTime:Y? might return -44.85, indicating Marker 1 (M1) is at the amplitude of -44.85 dBm.

## CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:X

Sets or queries the current bin number where the indicator is positioned from the histogram display. Sets or queries the frequency in Hz for the X indicator. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to FFT.

## Conditions

Measurement views: Pulse statistics

## Group

Calculate commands

## Syntax

CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:X <value>

CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:X?

## Related Commands

CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:Y?

## Arguments

<value>::=<NRf> specifies the indicator X value in Hz.

## Examples

CALCulate:PULSe:STATistics:MEASview2:FFT:INDicator:X 1000 sets the FFT indicator X value to 1 kHz.

## CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:Y? (Query )Only

Returns the frequency level in dB for the Y indicator. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to FFT.

## Conditions

Measurement views: Pulse statistics

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:Y?
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:FFT:INDicator:X
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> the indicator Y value in dB.

## Examples

CALCulate:PULSe:STATistics:MEASview2:FFT:INDicator:Y? might return -34.5 indicating that the FFT indicator is at -34.5 dB.

## CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:X

Sets or queries the histogram indicator X value. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to histogram.

## Conditions

Measurement views: Pulse statistics

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:X <value>
```

```
CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:X?
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<value>::=<NRf> specifies the indicator X value.

## Examples

CALCulate:PULSe:STATistics:MEASview2:HISTogram:INDicator:X 0.3 sets the histogram indicator X value to 0.3.

## CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:Y? (Query Only)

Queries the Histogram indicator Y value. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Histogram.

## Conditions

Measurement views: Pulse statistics

## Group

Calculate commands

## Syntax

CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:Y?

## Related Commands

CALCulate:PULSe:STATistics[:MEASview<y>]:HISTogram:INDicator:X

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> indicates the indicator Y value.

## Examples

CALCulate:PULSe:STATistics:MEASview2:HISTogram:INDicator:Y? might return 20 to indicate 20 counts in the histogram bin in dB.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:X?

Returns the delta marker time for the selected marker on the pulse trace. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Time Trend.



## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Related Commands

```
DISPlay:PULSe:STATistics:PLOT
```

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker time in seconds for the selected marker.

## Examples

CALCulate:PULSe:STATistics:MEASview1:MARKer1:DELTA:X? might return 3.01e-3, indicating that the delta marker time is 3.01 ms on the trace of pulse statistics measurement view 1.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:Y?

Returns the delta marker amplitude for the selected marker on the pulse trace. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

```
DISPlay:PULSe:STATistics:PLOT
```

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker amplitude for the selected marker, in the unit of the current measurement.

## Examples

CALCulate:PULSe:STATistics:MEASview1:MARKer1:DELTA:Y? might return 2.345, indicating that the delta marker amplitude is 2.345 dB on the trace of pulse statistics measurement view 1.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:MAXimum

Moves the selected marker to the highest peak on the pulse statistics trace. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:MAXimum

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:STATistics:MEASview1:MARKer1:MAXimum moves Marker 1 (M1) to the highest peak on the trace of pulse statistics measurement view 1.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:HIGHer

Moves the selected marker to the next peak higher in amplitude on the pulse statistics trace. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:STATistics:MEASview1:MARKer1:PEAK:HIGHer moves Marker 1 (M1) to the next peak higher in amplitude on the trace of pulse statistics measurement view 1.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LEFT

Moves the selected marker to the next peak to the left on the pulse statistics trace. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

`CALCulate:PULSe:STATistics:MEASview1:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace of pulse statistics measurement view 1.

## **CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LOWer**

Moves the selected marker to the next peak lower in amplitude on the pulse statistics trace. This command is valid when `DISPlay:PULSe:STATistics:PLOT` is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

`CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Related Commands

`CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Examples

`CALCulate:PULSe:STATistics:MEASview1:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace of pulse statistics measurement view 1.

## **CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:RIGHT**

Moves the selected marker to the next peak to the right on the pulse statistics trace. This command is valid when `DISPlay:PULSe:STATistics:PLOT` is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:STATistics[:MEASview1:MARKer1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace of pulse statistics measurement view 1.

## CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker in the pulse statistics view. This command is valid when DISPLAY:PULSe:STATistics:PLOT is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:X?
```

## Related Commands

```
CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:Y
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> specifies the horizontal position of the marker.

## Examples

`CALCulate:PULSe:STATistics:MEASview1:MARKer1:X 3.01e-3` places Marker 1 (M1) at 3.01 ms on the trace of Pulse Statistics Measurement View 1.

## **CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:Y?**

Queries the marker amplitude of the selected marker in the pulse statistics view. This command is valid when `DISPlay:PULSe:STATistics:PLOT` is set to Time Trend.

## Conditions

Measurement views: Pulse statistics with specified Measurement view number

## Group

Calculate commands

## Syntax

`CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:Y?`

## Related Commands

`CALCulate:PULSe:STATistics[:MEASview<y>]:MARKer<x>:X`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Returns

`<NRf>` Amplitude of the specified marker in the unit of the current measurement.

## Examples

`CALCulate:PULSe:STATistics:MEASview1:MARKer1:Y?` might return -28.173, indicating Marker 1 (M1) is at -28.173 dB on the trace of pulse statistics measurement view 1.

## **CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:X**

Returns the delta marker time for the selected marker on the pulse trace.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

CALCulate:PULSe:TRACe:MEASview1:MARKer1:DELTA:X? might return 38.0E-9, indicating that the delta marker time is 38.0 ns on the trace of pulse trace measurement view 1.

## CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:Y

Returns the delta marker amplitude for the selected marker on the pulse trace.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:DELTA:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

`CALCulate:PULSe:TRACe:MEASview1:MARKer1:DELta:Y?` might return 23.45, indicating that the delta marker amplitude is 23.45 dB on the trace of pulse trace measurement view 1.

## **CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:MAXimum**

Moves the selected marker to the highest peak on the pulse trace.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Calculate commands

### Syntax

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:MAXimum`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Examples

`CALCulate:PULSe:TRACe:MEASview1:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace of pulse trace measurement view 1.

## **CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:HIGHer**

Moves the selected marker to the next peak higher in amplitude on the pulse trace.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Calculate commands

### Syntax

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

### Related Commands

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LOWer`



## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:TRACe:MEASview1:MARKer1:PEAK:HIGHer moves Marker 1 (M1) to the next peak higher in amplitude on the trace of pulse trace measurement view 1.

## CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LEFT

Moves the selected marker to the next peak to the left on the pulse trace.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:TRACe:MEASview1:MARKer1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace of pulse trace measurement view 1.

## CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LOWer

Moves the selected marker to the next peak lower in amplitude on the pulse trace.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

CALCulate:PULSe:TRACe:MEASview1:MARKer1:PEAK:LOWer moves Marker 1 (M1) to the next peak lower in amplitude on the trace of pulse trace measurement view 1.

## CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:RIGHT

Moves the selected marker to the next peak to the right on the pulse trace.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Examples

`CALCulate:PULSe:TRACe:MEASview1:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the left on the trace of pulse trace measurement view 1.

## **CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker in the pulse trace view.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Calculate commands

### Syntax

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:X <value>`

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:X?`

### Related Commands

`CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:Y?`

### Arguments

`<value>::=<NRf>` specifies the horizontal position of the marker.



**Note:** To set the horizontal position, the marker must be on the trace of the pulse selected using the `DISPlay:PULSe:SElect:NUMBer` command. You cannot put the marker out of the horizontal range of the pulse.

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Returns

`<NRf>`

### Examples

`CALCulate:PULSe:TRACe:MEASview1:MARKer1:X 1.5`  $\mu$ s places Marker 1 (M1) at 1.5  $\mu$ s on the trace.

## **CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:Y**

Queries the marker amplitude of the selected marker in the pulse trace view.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:PULSe:TRACe[:MEASview<y>]:MARKer<x>:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

CALCulate:PULSe:TRACe:MEASview1:MARKer1:Y? might return -28.86, indicating Marker 1 (M1) is at -28.86 dBm on the trace of pulse trace measurement view 1.

## CALCulate:SEARch:LIMit:FAIL? (Query Only)

Queries whether the waveform cuts across the limit in the search operation.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:FAIL?
```

## Arguments

None

## Returns

{0|1}

0 represents Pass, indicating that the waveform does not cut across the limit.

1 represents Fail, indicating that the waveform cuts across the limit.

## Examples

`CALCULATE:SEARCH:LIMIT:FAIL?` might return 1, indicating that the waveform cuts across the limit (Fail).

## **CALCulate:SEARCh:LIMit:MATCh:BEEP[:STATe]**

Determines whether or not to beep on match during run or replay in the search operation.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

```
CALCulate:SEARCh:LIMit:MATCh:BEEP[:STATe] { OFF | ON | 0 | 1 }
```

```
CALCulate:SEARCh:LIMit:MATCh:BEEP[:STATe]?
```

### Arguments

OFF or 0 disables to beep on match.

ON or 1 enables to beep on match.

### Examples

```
CALCULATE:SEARCH:LIMIT:MATCH:BEEP:STATE
```

ON enables to beep on match.

## **CALCulate:SEARCh:LIMit:MATCh:SACQuire[:STATe]**

Determines whether or not to stop acquiring data on match during run or replay in the search operation.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

```
CALCulate:SEARCh:LIMit:MATCh:SACQuire[:STATe] { OFF | ON | 0 | 1 }
```

```
CALCulate:SEARCh:LIMit:MATCh:SACQuire[:STATe]?
```

### Arguments

OFF or 0 disables to stop acquiring data on match.

ON or 1 enables to stop acquiring data on match.

## Examples

`CALCULATE:SEARCH:LIMIT:MATCH:SACQUIRE:STATE ON` enables to stop acquiring data on match.

## **CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]**

Determines whether or not to save automatically (AutoSave) acquisition data on match during run in the search operation.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

`CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe] { OFF | ON | 0 | 1 }`

`CALCulate:SEARch:LIMit:MATCh:SDATa[:STATe]?`

## Arguments

OFF or 0 disables the AutoSave.

ON or 1 enables to save acquisition data automatically on match.

The data is saved to a file with the name:

`<name>-yyyy.mm.dd.hh.mm.ss.sss.tiq`

Where

`<name>` is the file name that was last specified.

`yyyy.mm.dd` and `hh.mm.ss.sss` represent date and 24 hour time.

The file extension is `.tiq`.

Example: `SAVED-2007.03.20.12.34.567.tiq`

For the directory of file, refer to *Specifying the File* (See page 2-89.) in the MMEMory command section.

## Examples

`CALCULATE:SEARCH:LIMIT:MATCH:SDATA:STATE ON` enables to save acquisition data automatically on match.

## **CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe]**

Determines whether or not to save automatically (AutoSave) the whole screen on match during run in the search operation.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe] { OFF | ON | 0 | 1 }
```

```
CALCulate:SEARch:LIMit:MATCh:SPICture[:STATe]?
```

## Arguments

OFF or 0 disables the AutoSave.

ON or 1 enables to save the whole screen automatically on match.

The picture is saved to a file with the name:

```
<name>-yyyy.mm.dd.hh.mm.ss.sss.png
```

Where

<name> is the file name that was last specified.

yyyy.mm.dd and hh.mm.ss.sss represent date and 24 hour time.

The file extension is .png.

Example: SAVED-2007.03.20.12.34.567.png

For the directory of file, refer to *Specifying the File* (See page 2-89.) in the

MMEMoRY command section.

## Examples

CALCULATE:SEARCH:LIMIT:MATCH:SPICTURE:STATE ON enables to save the whole screen automatically on match.

## CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]

Determines whether or not to save automatically (AutoSave) the spectrum trace on match during run in the search operation.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe] { OFF | ON | 0 | 1 }
```

```
CALCulate:SEARch:LIMit:MATCh:STRace[:STATe]?
```

## Arguments

OFF or 0 disables the AutoSave.

ON or 1 enables to save the spectrum trace automatically on match.

The trace is saved to a file with the name:

<name>-yyyy.mm.dd.hh.mm.ss.sss.Specan

Where

<name> is the file name that was last specified.

yyyy.mm.dd and hh.mm.ss.sss represent date and 24 hour time.

The file extension is .Specan.

Example: SAVED-2007.03.20.12.34.567.Specan

For the directory of file, refer to *Specifying the File* (See page 2-89.) in the MMEory command section.

## Examples

CALCULATE : SEARCH : LIMIT : MATCH : STRACE : STATE ON enables to save the spectrum trace automatically on match.

## CALCulate:SEARch:LIMit:OPERation

Sets or queries the search condition.

## Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

CALCulate:SEARch:LIMit:OPERation { LT | GT | IMASK | OMASK }

CALCulate:SEARch:LIMit:OPERation?

## Related Commands

CALCulate:[:SEARch]:TEST]:LIMit[:STATe]:MATCh:FILE:LOCation

## Arguments

**Table 31: The following table lists the arguments.**

Argument	Meaning
LT	The data is less than the limit value.
GT	The data is greater than the limit value.
IMASK	The data is inside the limit mask.

Table continued...



Argument	Meaning
OMASK	The data is outside the limit mask.

You can select the data using the CALCulate:SEARCh:LIMit:OPERation:FEED command.

You can set the limit value using the CALCulate[:SEARCh]:TEST[:LIMit[:STATe]:MATCh:FILE:LOCation command.

You can store and load the limit mask using the CALCulate:SEARCh:LIMit:OPERation:MASK:STORE and CALCulate:SEARCh:LIMit:OPERation:MASK:LOAD commands.

## Examples

CALCULATE:SEARCH:LIMIT:OPERATIONGT selects "the data is greater than the limit value" for the search condition.

## CALCulate:SEARCh:LIMit:OPERation:FEED

Sets or queries the data flow to be fed in the search operation.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

CALCulate:SEARCh:LIMit:OPERation:FEED <view>,<trace>

CALCulate:SEARCh:LIMit:OPERation:FEED?

### Arguments

<view>::=<string> and <trace>::=<string> are listed in the following table.

Table 32: Source data of the search operation

<view>	<trace>	Meaning
"Spectrum"	"Trace 1" <sup>1</sup>	Trace 1 in the Spectrum view.
	"Trace 2" <sup>1</sup>	Trace 2 in the Spectrum view.
	"Trace 3" <sup>1</sup>	Trace 3 in the Spectrum view.
	"Math Trace"	Math trace in the Spectrum view.
	"Spectrogram Trace"	Spectrogram trace in the Spectrum view.

<sup>1</sup> There is a space character between Trace and the number.

## Examples

CALCULATE:SEARCH:LIMIT:OPERATION:FEED"Spectrum","Trace 1" selects the Trace 1 in the Spectrum measurement view for the search operation.

CALCULATE:SEARCH:LIMIT:OPERATION:FEED?

might return "Spectrum", "Math Trace", indicating that the math trace is used as the source data in the search operation.

## **CALCulate:SEARch:LIMit:OPERation:MASK:LOAD (No Query Form)**

Loads the specified limit mask file for the search operation.

### **Conditions**

Measurement views: All

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEARch:LIMit:OPERation:MASK:LOAD <file_name>
```

### **Arguments**

<file\_name>::=<string> specifies the file to load the limit mask from. The file extension is .lmt. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-63.) in the MMEemory command section.

### **Examples**

```
CALCULATE:SEARCH:LIMIT:OPERATION:MASK:LOAD"Limit1"
```

loads the limit mask from the Limit1.lmt file.

## **CALCulate:SEARch:LIMit:OPERation:MASK:STORE (No Query Form)**

Stores the limit mask to a specified file in the search operation.

### **Conditions**

Measurement views: All

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEARch:LIMit:OPERation:MASK:STORE <file_name>
```

### **Arguments**

<file\_name>::=<string> specifies the file to store the limit mask to. The file extension is .lmt. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-63.) in the MMEemory command section.

## Examples

`CALCULATE:SEARCH:LIMIT:OPERATION:MASK:STORE"Limit1"` stores the limit mask to the `Limit1.lmt` file.

## CALCulate:SEARch:LIMit:OPERation:SLIMit

Sets or queries the limit value (in dB) in the search operation.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:OPERation:SLIMit <value>`

`CALCulate:SEARch:LIMit:OPERation:SLIMit?`

### Related Commands

`CALCulate:SEARch:LIMit:OPERation`

### Arguments

`<value>::=<NRf>` specifies the limit value in the search operation. Range: -100 to +100 dBm.

## Examples

`CALCULATE:SEARCH:LIMIT:OPERATION:SLIMIT`

-20 sets the limit value to -20 dBm.

## CALCulate:SEARch:LIMit:REPort:DATA? (Query Only)

Returns the frequency range(s) that satisfy the search condition.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:REPort:DATA?`

### Arguments

None

## Returns

`<num_range>`, `<range(1)>`, `<range(2)>`, ..., `<range(n)>`

Where

`<num_range>` ::= `<NR1>` is the number of ranges that satisfy the condition.

`<range(n)>` ::= "`<lower_freq(n)>`, `<upper_freq(n)>`" (string)

represents the *n*th frequency range that satisfy the search condition in ascending order.

`<lower_freq(n)>` and `<upper_freq(n)>` are the lower and upper frequencies of the range #*n*, respectively.

## Examples

`CALCULATE:SEARCH:LIMIT:REPORT:DATA?` might return `2,"1.4800E+9,1.5001E+9","1.5002E+9,1.5200E+9"`, indicating that the search condition is satisfied in these two ranges 1.48 to 1.5001 GHz and 1.5002 to 1.52 GHz.

## CALCulate:SEARch:LIMit:REPort:POINts? (Query Only)

Returns the number of frequency range(s) that satisfy the search condition.

### Conditions

Measurement views: All

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:REPort:POINts?`

### Arguments

None

### Returns

`<number>` ::= `<NR1>` represents the number of frequency range(s) that satisfy the search condition.

## Examples

`CALCULATE:SEARCH:LIMIT:REPORT:POINTS?` might return `5`, indicating that five ranges satisfy the search condition.

## CALCulate:SEARch:LIMit:STATe

Sets or queries the search function (enabled or disabled).

### Conditions

Measurement views: All

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe { OFF | ON | 0 | 1 }
```

```
CALCulate:SEARch:LIMit:STATe?
```

## Arguments

OFF or 0 disables the search function.

ON or 1 enables the search function.

## Examples

```
CALCULATE:SEARCH:LIMIT:STATEON enables the search function.
```

## CALCulate:SEARch:LIMit:STATe:MATCH:ACQuisition:FORMat

Sets and queries the format in which the acquisition data to be saved in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:MATCH:ACQuisition:FORMat {TIQ | CSV | MAT}
```

```
CALCulate:SEARch:LIMit:STATe:MATCH:ACQuisition:FORMat?
```

## Arguments

TIQ saves acquisition data in `.Tiq` file format.

CSV saves acquisition data in `.csv` file format.

MAT saves acquisition data in `.Mat` file format.

## Returns

<String> Either TIQ, CSV or MAT

## Examples

```
CALCULATE:SEARCH:LIMIT:STATE:MATCH:ACQUISITION:FORMAT might return CSV, indicating that the acquisition data is saved in .csv file format.
```

## **CALCulate:SEARch:LIMit:STATe:MATCh:BEEP:STATe**

Determines the enabling and disabling of Beep check box in mask test control settings.

### **Conditions**

Measurement view: Spectrum

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEARch:LIMit:STATe:MATCh:BEEP:STATe {0 | OFF | 1 | ON}
```

```
CALCulate:SEARch:LIMit:STATe:MATCh:BEEP:STATe?
```

### **Arguments**

0 or OFF disables the Beep check box

1 or ON enables the Beep check box

### **Returns**

<Boolean> 0,1

### **Examples**

CALCULATE:SEARCH:LIMIT:STATE:MATCH:BEEP:STATE might return 0, indicating that the Beep check box is disabled.

## **CALCulate:SEARch:LIMit:STATe:MATCh:COUNT:STATe**

Determines the enabling and disabling of Max files check box in mask test control settings.

### **Conditions**

Measurement view: Spectrum

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEARch:LIMit:STATe:MATCh:COUNT:STATe {0 | OFF | 1 | ON}
```

```
CALCulate:SEARch:LIMit:STATe:MATCh:COUNT:STATe?
```

### **Arguments**

0 or OFF disables the Max files check box

1 or ON enables the Max files check box

## Returns

<Boolean> 0,1

## Examples

CALCULATE:SEARCH:LIMIT:STATE:MATCH:COUNT:STATE might return 0 indicating that the Max files check box is disabled.

## CALCulate:SEARCh:LIMit:STATe:MATCh:COUNt:STATe:MAXimum

Sets or queries the Max files value in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

CALCulate:SEARCh:LIMit:STATe:MATCh:COUNt:STATe:MAXimum<value>

CALCulate:SEARCh:LIMit:STATe:MATCh:COUNt:STATe:MAXimum?

## Arguments

<value> = <NR1> Range is 1 to 10000

## Returns

<NR1>

## Examples

CALCULATE:SEARCH:LIMIT:STATE:MATCH:COUNT:STATE:MAXIMUM might return 250 indicating the value of max files given is 250.

## CALCulate:SEARCh:LIMit:STATe:MATCh:FILE:LOCation

Sets or queries the location of the file in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

CALCulate:SEARCh:LIMit:STATe:MATCh:FILE:LOCation <file\_name>

CALCulate:SEARch:LIMit:STATe:MATCh:FILE:LOCation?

### Arguments

<file\_name> = <String> Specifies the location of the file

### Returns

<String>

### Examples

CALCULATE:SEARCH:LIMIT:STATE:MATCH:FILE:LOCATION might return "C:\SignalVu-PC Files", indicating the location of the file is "C:\SignalVu-PC Files".

## CALCulate:SEARch:LIMit:STATe:MATCh:FILE:NAME:BASE

Sets and queries the file name in mask test control settings.

### Conditions

Measurement view: Spectrum

### Group

Calculate commands

### Syntax

CALCulate:SEARch:LIMit:STATe:MATCh:FILE:NAME:BASE <file\_name>

CALCulate:SEARch:LIMit:STATe:MATCh:FILE:NAME:BASE?

### Arguments

<file\_name> = <String> Specifies the name of the file

### Returns

<String>

### Examples

CALCULATE:SEARCH:LIMIT:STATE:MATCH:FILE:NAME:BASE might return "Sample1", indicating the name of the file is "Sample1".

## CALCulate:SEARch:LIMit:STATe:MATCh:PICTure:FORMat

Sets and queries the format in which the picture to be saved in mask test control settings.

### Conditions

Measurement view: Spectrum



## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:MATCH:PICTure:FORMat {PNG | JPG | BMP}
```

```
CALCulate:SEARch:LIMit:STATe:MATCH:PICTure:FORMat ?
```

## Arguments

PNG saves picture in png format.

JPG saves picture in jpg format.

BMP saves picture in bmp format.

## Returns

<String> Either PNG, JPG or BMP

## Examples

CALCULATE:SEARCH:LIMIT:STATE:MATCH:PICTURE:FORMAT might return JPG, indicating that the Picture is saved in jpg format.

## CALCulate:SEARch:LIMit:STATe:MATCH:SACQuire:STATe

Determines the enabling and disabling of Stop check box in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:MATCH:SACQuire:STATe {0 | OFF | 1 | ON}
```

```
CALCulate:SEARch:LIMit:STATe:MATCH:SACQuire:STATe?
```

## Arguments

0 or OFF disables the stop check box

1 or ON enables the stop check box

## Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:MATCH:SACQUIRE:STATE` might return 1, indicating that the stop check box is enabled.

## **CALCulate:SEARch:LIMit:STATe:MATCh:SDATa:STATe**

Determines the enabling and disabling of Save acquisition data check box in mask test control settings.

### Conditions

Measurement view: Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:STATe:MATCh:SDATa:STATe {0 | OFF | 1 | ON}`

`CALCulate:SEARch:LIMit:STATe:MATCh:SDATa:STATe ?`

### Arguments

0 or OFF disables the Save acquisition data check box

1 or ON enables the Save acquisition data check box

### Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:MATCH:SDATA:STATE` might return 1, indicating that the Save acquisition data check box is enabled.

## **CALCulate:SEARch:LIMit:STATe:MATCh:SPICture:STATe**

Determines the enabling and disabling of Save picture check box in mask test control settings.

### Conditions

Measurement view: Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:STATe:MATCh:SPICture:STATe {0 | OFF | 1 | ON}`

`CALCulate:SEARch:LIMit:STATe:MATCh:SPICture:STATe?`

## Arguments

0 or OFF disables the Save picture check box

1 or ON enables the Save picture check box

## Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:MATCH:SPICTURE:STATE` might return 1, indicating that the Save picture check box is enabled.

## **CALCulate:SEARCh:LIMit:STATe:MATCH:STRace:STATe**

Determines the enabling and disabling of Save trace check box in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

`CALCulate:SEARCh:LIMit:STATe:MATCH:STRace:STATe {0 | OFF | 1 | ON}`

`CALCulate:SEARCh:LIMit:STATe:MATCH:STRace:STATe ?`

## Arguments

0 or OFF disables the Save trace check box

1 or ON enables the Save trace check box

## Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:MATCH:STRACE:STATE` might return 0, indicating that the Save trace check box is disabled.

## **CALCulate:SEARCh:LIMit:STATe:OPERation**

Sets or queries the region in which the test should be performed in mask test xcontrol settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:OPERation {GT | LT | OMAS | IMAS}
```

```
CALCulate:SEARch:LIMit:STATe:OPERation ?
```

## Arguments

GT selects “Is greater than” as the region for the test to be performed

LT selects “Is less than” as the region for the test to be performed

OMAS selects “Is Outside mask” as the region for the test to be performed

IMAS selects “Is inside mask” as the region for the test to be performed

## Returns

<String> Either GT , LT , OMAS or IMAS

## Examples

CALCULATE:SEARCH:LIMIT:STATE:OPERATION might return LT, indicating “Is less than” is selected as the region for the test to be performed.

## CALCulate:SEARch:LIMit:STATe:OPERation:FEED

Sets and queries the trace to which the test should be performed in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:OPERation:FEED <Arg>
```

```
CALCulate:SEARch:LIMit:STATe:OPERation:FEED?
```

## Arguments

<Arg> =

“Spectrum”, “Trace 1” selects Trace 1 in spectrum

“Spectrum”, “Trace 2” selects Trace 2 in spectrum

“Spectrum”, “Trace 3” selects Trace 3 in spectrum

“Spectrum”, “Math Trace” selects math trace in spectrum

“Spectrum”, “Spectrogram Trace” selects Spectrogram trace in spectrum

## Returns

<String> See Arguments

## Examples

CALCULATE:SEARCH:LIMIT:STATE:OPERATION:FEED might return "Spectrum","Trace 1", indicating that the test should be performed on "Spectrum","Trace 1".

## CALCulate:SEARch:LIMit:STATe:OPERation:MASK:LOAD (No Query Form)

Sets the location of the file which has to be loaded in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:OPERation:MASK:LOAD <file_name>
```

## Arguments

<file\_name> = <String> Specifies the location of the file

## Returns

None

## Examples

CALCULATE:SEARCH:LIMIT:STATE:OPERATION:MASK:LOAD "C:\SignalVu-PC Files" loads the data from the file "C:\SignalVu-PC Files".

## CALCulate:SEARch:LIMit:STATe:OPERation:MASK:LOWer:STATe

Determines the enabling and disabling of Lower mask check box in mask test editor tab of mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe:OPERation:MASK:LOWer:STATe {0 | OFF | 1 | ON}
```

```
CALCulate:SEARch:LIMit:STATe:OPERation:MASK:LOWer:STATe?
```

### Arguments

0 or OFF disables the Lower mask check box

1 or ON enables the Lower mask check box

### Returns

<Boolean> 0,1

### Examples

CALCULATE:SEARCH:LIMIT:STATE:OPERATION:MASK:LOWER:STATE might return 0, indicating that the Lower mask check box is disabled.

## CALCulate:SEARch:LIMit:STATe:OPERation:MASK:NEW:AUTO (No Query Form)

Draws a new frequency mask automatically based on a reference trace.

### Conditions

Measurement view: Spectrum, DPX, Spectrogram

### Group

Calculate commands

### Syntax

CALCulate:SEARch:LIMit:STATe:OPERation:MASK:NEW:AUTO

### Arguments

<x\_margin>::<NRf> specifies the X margin (horizontal offset from the reference trace) in Hz.

<y\_margin>::<NRf> specifies the Y margin (vertical offset from the reference trace) in dB.

<meas\_ID>::<string> specifies the measurement view.

]<trace\_ID> specifies the reference trace.

The values for the above arguments are listed in the following table for each possible measurement view.

Measurement view	<meas_ID>	<trace_ID>
Spectrum	"specan"	TRACE1 (Trace 1)
		TRACE2 (Trace 2)
		TRACE3 (Trace 3)
		TRACE4 (Math trace)
		TRACE5 (Spectrogram trace)
DPX, Spectrogram	DPX	TRACE1 (+Peak trace)
		TRACE2 (-Peak trace)
		TRACE3 (Avg trace)
		TRACE4 (Math trace)

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:OPERATION:MASK:NEW:AUTO" specan",TRACE1,7E+6,5` automatically draws a new frequency mask in the Spectrum view based on Trace 1 with a horizontal margin of 7 MHz and a vertical margin of 5 dB

## **CALCulate:SEARch:LIMit:STATe:OPERation:MASK:STORE (No Query Form)**

Sets the location of the file to which data has to be saved in mask test control settings.

### Conditions

Measurement view: Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:STATe:OPERation:MASK:STORE <file_name>`

### Arguments

`<file_name>` = `<String>` Specifies the location of the file

### Returns

None

### Examples

`CALCULATE:SEARCH:LIMIT:STATE:OPERATION:MASK:STORE`

`"C:\SignalVu-PC Files"` saves the data to the file `"C:\SignalVu-PC Files"`.

## **CALCulate:SEARch:LIMit:STATe:OPERation:MASK:UPPer:STATe**

Determines the enabling and disabling of Upper mask check box in mask test editor tab of mask test control settings.

### Conditions

Measurement view: Spectrum

### Group

Calculate commands

### Syntax

`CALCulate:SEARch:LIMit:STATe:OPERation:MASK:UPPer:STATe {0 | OFF | 1 | ON}`

`CALCulate:SEARch:LIMit:STATe:OPERation:MASK:UPPer:STATe?`

## Arguments

0 or OFF disables the Upper mask check box

1 or ON enables the Upper mask check box

## Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE:OPERATION:MASK:UPPER:STATE` might return 0, indicating that the Upper mask check box is disabled.

## CALCulate:SEARch:LIMit:STATe

Determines the enabling and disabling of Enable test check box in mask test control settings.

## Conditions

Measurement view: Spectrum

## Group

Calculate commands

## Syntax

```
CALCulate:SEARch:LIMit:STATe {0 | OFF | 1 | ON}
```

```
CALCulate:SEARch:LIMit:STATe?
```

## Arguments

0 or OFF disables the Enable test check box

1 or ON enables the Enable test check box

## Returns

<Boolean> 0,1

## Examples

`CALCULATE:SEARCH:LIMIT:STATE` might return 1, indicating that the Enable test check box is enabled.

## CALCulate:SEM:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the selected marker on the Spectral Emissions Mask trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.



## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

```
CALCulate:SEM:MARKer<x>:DELTA:X?
```

## Arguments

None

## Returns

<NRf>Delta marker frequency for the selected marker.

## Examples

CALCulate:SEM:MARKer1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

## CALCulate:SEM:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the Spectral Emissions Mask trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

```
CALCulate:SEM:MARKer<x>:DELTA:Y?
```

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCulate:SEM:MARKer1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## **CALCulate:SEM:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the Spectral Emissions Mask trace.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEM:MARKer<x>:MAXimum
```

### **Arguments**

None

### **Examples**

`CALCulate:SEM:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:SEM:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the Spectral Emissions Mask trace.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SEM:MARKer<x>:PEAK:HIGHer
```

### **Arguments**

None

### **Examples**

`CALCulate:SEM:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:SEM:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the Spectral Emissions Mask trace.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

```
CALCulate:SEM:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCulate:SEM:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## CALCulate:SEM:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on the Spectral Emissions Mask trace.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

```
CALCulate:SEM:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALCulate:SEM:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## CALCulate:SEM:MARKer<x>:PEAK:RIGHT (No Query Form)

Moves the selected marker to the next peak to the right on the Spectral Emissions Mask trace.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

`CALCulate:SEM:MARKer<x>:PEAK:RIGHT`

## Arguments

None

## Examples

`CALCulate:SEM:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:SEM:MARKer<x>[:SET]:CENTER (No Query Form)**

Moves the specified marker to the center frequency.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

`CALCulate:SEM:MARKer<x>[:SET]:CENTER`

## Arguments

None

## Examples

`CALCulate:SEM:MARKer1[:SET]:CENTER` moves Marker 1 (M1) to the center frequency.

## **CALCulate:SEM:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker on the Spectral Emissions Mask trace.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

`CALCulate:SEM:MARKer<x>:X <value>`

## Arguments

<value> ::= <NRf> specifies the horizontal position of the marker.

## Examples

CALCULATE:SEM:MARKER1:X 800MHz places Marker 1 (M1) at 800 MHz on the spectrum trace.

## CALCulate:SEM:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker on the Spectral Emissions Mask trace.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Calculate commands

## Syntax

CALCulate:SEM:MARKer<x>:Y?

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

CALCULATE:SEM:MARKER1:Y? might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X:FREQuency

Returns the delta marker frequency for the selected marker in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands

## Syntax

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X:FREQuency?

## Related Commands

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:Y?

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf>Delta marker frequency for the selected marker.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:DELTA:X:FREQUENCY? might return 5.95E+6, indicating that the delta marker frequency is 5.95 MHz.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME] (Query Only)

Returns the delta marker time for the selected marker in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands

## Syntax

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?

## Related Commands

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:Y?

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf>Delta marker time for the selected marker.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:DELTA:X:TIME? might return -1.84E-3, indicating that the delta marker time is -1.84 ms.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:Y (Query Only)

Returns the delta marker amplitude for the selected marker in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Calculate commands

### Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

### Related Commands

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:DELTA:X:TIME
```

### Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

<NRf>Delta marker amplitude for the selected marker.

### Examples

CALCulate:SGRam:MEASview1:MARKer1:DELTA:Y? might return -8.45, indicating that the delta marker amplitude is -8.45 dB.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)

Moves the selected marker to the highest peak on a line in the spectrogram. The line is selected using the TRACe:SGRam:SElect:LINE command.

### Conditions

Measurement views: Spectrogram

### Group

Calculate commands

### Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:MAXimum
```

### Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:MAXimum moves Marker 1 (M1) to the highest peak on the line.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on a line in the spectrogram. The line is selected using the TRACe:SGRam:SElect:LINE command.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands

## Syntax

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:HIGHer

## Related Commands

CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LOWer

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:PEAK:HIGHer moves Marker 1 (M1) to the next peak higher in amplitude on the line.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on a line in the spectrogram. The line is selected using the TRACe:SGRam:SElect:LINE command.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands



## Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:PEAK:LEFT moves Marker 1 (M1) to the next peak to the left on the trace of Spectrum Measurement View 1.

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)

Moves the selected marker to the next peak lower in amplitude on a line in the spectrogram. The line is selected using the TRACe:SGRam:SELEct:LINEcommand.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands

## Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

CALCulate:SGRam:MEASview1:MARKer1:PEAK:LOWer moves Marker 1 (M1) to the next peak lower in amplitude on the line.

## **CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on a line in the spectrogram. The line is selected using the TRACe:SGRam:SElect:LINE command.

### **Conditions**

Measurement views: Spectrogram

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```

### **Related Commands**

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

### **Arguments**

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Examples**

CALCulate:SGRam:MEASview1:MARKer1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace of Spectrogram Measurement View 1.

## **CALCulate:SGRam[:MEASview<y>]:MARKer<x>[:SET]:CENTER (No Query Form)**

Sets the center frequency to the marker frequency in the spectrogram.

### **Conditions**

Measurement views: Spectrogram

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>[:SET]:CENTER
```

### **Arguments**

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

`CALCulate:SGRam:MEASview1:MARKer1:SET:CENTER` sets the center frequency to the Marker 1 frequency in the spectrogram.

## **CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X:FREQuency**

Sets or queries the marker frequency in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Calculate commands

### Syntax

`CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X:FREQuency <value>`

`CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X:FREQuency?`

### Related Commands

`CALCulate:SGRam[:MEASview<x>]:MARKer<y>:Y?`

### Arguments

`<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command

`<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument. where `MEASview` keyword is optional by default `y` is 1.

`<value>::=<NRf>` specifies the marker frequency. Range: (center frequency)  $\pm$  (span)/2.

### Returns

`<NRf>` specifies the marker frequency.

### Examples

`CALCulate:SGRam:MEASview1:MARKer1:X:FREQuency 800MHz` places Marker 1 (M1) at 800 MHz on the trace.

## **CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X[:TIME]**

Sets or queries the marker time in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Calculate commands

## Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X[:TIME] <value>
```

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:X[:TIME]?
```

## Related Commands

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:Y?
```

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>: :=<NRf> specifies the marker time.

## Returns

<NRf> specifies the marker time.

## Examples

```
CALCulate:SGRam:MEASview1:MARKer1:X:TIME -234.5us places Marker 1 (M1) at -234.5 µs on the trace.
```

## CALCulate:SGRam[:MEASview<y>]:MARKer<x>:Y (Query Only)

Queries the marker amplitude in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Calculate commands

## Syntax

```
CALCulate:SGRam[:MEASview<y>]:MARKer<x>:Y?
```

## Related Commands

```
CALCulate:SGRam[:MEASview<x>]:MARKer<y>:X:TIME
```

## Arguments

<x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> Marker amplitude of the selected marker.

## Examples

`CALCulate:SGRam:MEASview1:MARKer1:Y?` might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the selected marker on the spectrum trace.

### Conditions

Measurement views: Spectrum with the specified Measurement View Number.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

### Group

Calculate commands

### Syntax

`CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:X? (QUERY)`

### Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

<NRf> Delta marker frequency for the selected marker.

## Examples

`CALCulate:SPECTrum:MEASview1:MARKer1:DELTA:X?` might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the spectrum trace.

### Conditions

Measurement views: Spectrum with the specified Measurement View Number

### Group

Calculate commands

### Syntax

`CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:Y? (QUERY)`

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:MAXimum

Moves the selected marker to the highest peak on the spectrum trace.

## Conditions

Measurement view: Spectrum with the specified Measurement View Number

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:MAXimum moves Marker 1 (M1) to the highest peak on the trace of Spectrum Measurement View 1.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

## Conditions

Measurement views: Spectrum with the specified Measurement View Number.

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:PEAK:HIGHer moves Marker 1 (M1) to the next higher peak in amplitude on the trace of Spectrum Measurement View 1

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the selected marker to the next peak to the left on the spectrum trace.

## Conditions

Measurement views: Spectrum with the specified Measurement View Number.

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

`CALCulate:SPECTrum:MEASview1:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the right on the trace of Spectrum Measurement View 1.

## **CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the spectrum trace.

### Conditions

Measurement views: Spectrum with the specified Measurement View Number.

### Group

Calculate commands.

### Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

### Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

<NRf>

## Examples

`CALCulate:SPECTrum:MEASview1:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next lower peak in amplitude on the trace of Spectrum Measurement View 1.

## **CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the spectrum trace.

### Conditions

Measurement views: Spectrum with the specified Measurement View Number

### Group

Calculate commands

### Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:PEAK:RIGHT
```



## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:PEAK:RIGHT moves Marker 1 (M1) to the next peak to the right on the trace of Spectrum Measurement View 1.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:DENSITY? (Query Only)

Returns the power density of the specified marker in the Spectrum measurement.

## Conditions

Measurement view: Spectrum with the specified Measurement View Number

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Group

Calculate commands

## Syntax

CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:DENSITY? (QUERY)

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NR2> = a floating point value without exponents. It is the power density of the selected marker in dBm/Hz.

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:POWER:DENSITY? might return -80.2412563552, indicating that the power density is -80.24 dBm/Hz.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWER:IDENSITY?

Returns the integrated power density reading between the MR (reference marker) and the selected marker in the Spectrum measurement.

## Conditions

Measurement view: Spectrum with the specified Measurement View Number.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWer:IDENsity? <QUERY>
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NR2> = a floating point value without exponents. It is the integrated power density reading between MR (reference marker) and the selected marker in dBm/Hz.

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:POWer:IDENsity? might return -78.5625446112, indicating that the integrated power density is -78.56 dBm/Hz.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWer:INTEgrated? (Query Only)

Returns the integrated power reading between the MR (reference marker) and the selected marker in the Spectrum measurement.

## Conditions

Measurement view: Spectrum with the specified Measurement View Number.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:POWer:INTEgrated? (QUERY)
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NR2> = a floating point value without exponents. It is the integrated power reading between MR (reference marker) and the selected marker in dBm.

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:POWer:INTEgrated? might return -12.4462022781, indicating the integrated power is -12.45 dBm.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>[:SET]:CENTer (No Query Form)

Sets the center frequency to the marker frequency in the spectrum measurement.

## Conditions

Measurement views: Spectrum with the specified Measurement View Number

## Group

Calculate commands

## Syntax

CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>[:SET]:CENTer

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:SET:CENTer sets the center frequency to the marker frequency in the Spectrum Measurement View 1

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:TRACe

Sets or queries the trace on which the specified marker is placed in the spectrum measurement.

## Conditions

Measurement views: Spectrum with specified Measurement View

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3 | TRACE4 | TRACE5 }
```

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:TRACe? (QUERY)
```

## Arguments

TRACE1 places the specified marker on Trace 1.

TRACE2 places the specified marker on Trace 2.

TRACE3 places the specified marker on n Trace 3.

TRACE4 places the specified marker on Trace 4 (math trace).

TRACE5 places the specified marker on Trace 5 (spectrogram).

Trace 1 to 3 can be defined as Normal, Average, Max Hold or Min Hold using the command.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

```
CALCulate:SPECTrum:MEASview1:MARKer1:TRACe1 places Marker 1 (M1) on Trace 1 of Spectrum Measurement View
```

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:X

Sets or queries the horizontal position of the selected marker on the spectrum trace.

## Conditions

Measurement views: Spectrum with the specified Measurement View Number.

## Group

Calculate commands

## Syntax

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:X? <QUERY>
```

## Arguments

<value>

Range : Start to Stop frequency (left to right edge of the horizontal axis).

Using an out-of-range value causes an execution error (-222, "Data out of range").

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:X 1GHz places Marker 1 (M1) at 1GHz on the trace of Spectrum Measurement View 1.

## CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the spectrum trace

## Conditions

Measurement views: Spectrum with the specified Measurement View Number

## Group

Calculate commands

## Syntax

CALCulate:SPECTrum[:MEASview<y>]:MARKer<x>:DELTA:Y? (QUERY)

## Related Commands

[CALCulate:SPECTrum:MARKer<x>:X](#)

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

CALCulate:SPECTrum:MEASview1:MARKer1:DELTA:Y? might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## **CALCulate:SPURious:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker frequency for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement views: Spurious

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SPURious:MARKer<x>:DELTA:X?
```

### **Arguments**

None

### **Returns**

<NRf> Delta marker frequency for the selected marker.

### **Examples**

CALCULATE:SPURIOUS:MARKER1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

## **CALCulate:SPURious:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the spectrum trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### **Conditions**

Measurement views: Spurious

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:SPURious:MARKer<x>:DELTA:Y?
```

### **Arguments**

None

### **Returns**

<NRf> Delta marker amplitude for the selected marker.

## Examples

`CALCULATE:SPURIOUS:MARKER1:DELTA:Y?` might return 23.45, indicating that the delta marker amplitude is 23.45 dB.

## **CALCulate:SPURious:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the spectrum trace.

### Conditions

Measurement views: Spurious

### Group

Calculate commands

### Syntax

`CALCulate:SPURious:MARKer<x>:MAXimum`

### Arguments

None

## Examples

`CALCULATE:SPURIOUS:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:SPURious:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the spectrum trace.

### Conditions

Measurement views: Spurious

### Group

Calculate commands

### Syntax

`CALCulate:SPURious:MARKer<x>:PEAK:HIGHer`

### Arguments

None

## Examples

`CALCULATE:SPURIOUS:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:SPURious:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the spectrum trace.

### **Conditions**

Measurement views: Spurious

### **Group**

Calculate commands

### **Syntax**

`CALCulate:SPURious:MARKer<x>:PEAK:LEFT`

### **Arguments**

None

### **Examples**

`CALCULATE:SPURIOUS:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:SPURious:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the spectrum trace.

### **Conditions**

Measurement views: Spurious

### **Group**

Calculate commands

### **Syntax**

`CALCulate:SPURious:MARKer<x>:PEAK:LOWer`

### **Arguments**

None

### **Examples**

`CALCULATE:SPURIOUS:MARKER1:PEAK:LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:SPURious:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the spectrum trace.

### **Conditions**

Measurement views: Spurious



## Group

Calculate commands

## Syntax

```
CALCulate:SPURious:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCULATE:SPURIOUS:MARKER1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:SPURious:MARKer<x>[:SET]:CENTER (No Query Form)**

Sets the center frequency to the marker frequency in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Calculate commands

## Syntax

```
CALCulate:SPURious:MARKer<x>[:SET]:CENTer
```

## Arguments

None

## Examples

`CALCULATE:SPURIOUS:MARKER1:SET:CENTER` sets the center frequency to the value at Marker 1.

## **CALCulate:SPURious:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker on the spectrum trace.

## Conditions

Measurement views: Spurious

## Group

Calculate commands

## Syntax

```
CALCulate:SPURious:MARKer<x>:X <value> CALCulate:SPURious:MARKer<x>:X?
```

## Arguments

`<value> ::= <NRf>` specifies the horizontal position of the marker.

Range: Start to Stop frequency (left to right edge of the horizontal axis).

The start and stop frequencies are set using the `DISPlay:SPURious:X[:SCALe]:STARt` and `DISPlay:SPURious:X[:SCALe]:STOP` commands.

## Examples

`CALCULATE:SPURIOUS:MARKER1:X 800MHz` places Marker 1 (M1) at 800 MHz on the spectrum trace.

## CALCulate:SPURious:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker on the spectrum trace.

## Conditions

Measurement views: Spurious

## Group

Calculate commands

## Syntax

`CALCulate:SPURious:MARKer<x>:Y?`

## Arguments

None

## Returns

`<NRf>` Marker amplitude of the selected marker.

## Examples

`CALCULATE:SPURIOUS:MARKER1:Y?` might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.

## CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]? (Query only)

Returns the delta marker time for the selected marker on the Trellis diagram trace.

The parameter `<x>` = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

`MEASview` keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:DELTA:X[:TIME]?
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:DELTA:Y?](#)

## Arguments

None

## Returns

<NRf> Delta marker time for the selected marker.

Use the [SENSE]:DDEMod:TIME:UNITS command to select the time unit: symbols (default) or seconds.

## Examples

CALCulate:TDIagram :MEASview2:MARKer1:DELTA:X:TIME? might return 62.75, indicating that the delta marker time is 62.75 symbols.

## CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:DELTA:Y? (Query only)

Returns the delta marker amplitude for the selected marker on the Trellis diagram trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using CALCulate:MARKer:ADD command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:DELTA:Y?
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:DELTA:X\[:TIME\]?](#)

## Arguments

None

## Returns

<NRf> Delta marker amplitude for the selected marker in degrees.

## Examples

`CALCulate:TDIagram:MEASview2:MARKer1:DELTA:Y?` might return -48.26, indicating that the delta marker amplitude is -48.26°.

## **CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:MAXimum (No query form)**

Moves the selected marker to the highest peak on the Trellis diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:TDIagram:MEASview1:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No query form)**

Moves the selected marker to the next peak higher in amplitude on the Trellis diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram)

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer](#)

## Arguments

None

## Examples

`CALCulate:TDIagram:MEASview2:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No query form)**

Moves the selected marker to the next peak to the left on the Trellis diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCulate:TDiagram:MEASview2:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No query form)**

Moves the selected marker to the next peak lower in amplitude on the Trellis diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

`CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:LOWer`

## Related Commands

[\*CALCulate:TDiagram\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer\*](#)

## Arguments

None

## Examples

`CALCulate:TDiagram:MEASview2:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No query form)**

Moves the selected marker to the next peak to the right on the Trellis diagram trace.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>] :MARKer<x>:PEAK:RIGHT
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

None

## Examples

`CALCulate:TDIagram:MEASview2:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:X[:TIME]

Sets or queries the horizontal position (time) of the selected marker in the Trellis diagram measurement.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

```
CALCulate:TDIagram[:MEASview<y>]:MARKer<x>:X[:TIME] <value>
```

```
CALCulate:TDIagram:MARKer<x>:X[:TIME] ?
```

## Related Commands

[CALCulate:TDIagram\[:MEASview<y>\]:MARKer<x>:Y?](#)

## Arguments

<value>::=<NRf> specifies the horizontal position (time) of the marker. Use the `[SENSe]:DDEMod:TIME:UNITs` command to select the time unit as symbols (default) or seconds.

## Examples

`CALCulate:TDiagram :MEASview2:MARKer1:X:TIME 38.5` places Marker 1 (M1) at 38.5 symbols on the trace.

## **CALCulate:TDiagram [:MEASview<y>]:MARKer<x>:Y? (Query only)**

Queries the vertical position of the selected marker in the Trellis diagram measurement.

The parameter <x> = 0 to 4. The specified marker must be activated using `CALCulate:MARKer:ADD` command.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Calculate commands

## Syntax

`CALCulate:TDiagram[:MEASview<y>]:MARKer<x>:Y?`

## Related Commands

[CALCulate:TDiagram\[:MEASview<y>\]:MARKer<x>:X\[:TIME\]](#)

## Arguments

None

## Returns

<NRf> Marker amplitude of the selected marker in degrees.

## Examples

`CALCulate:TDiagram:MEASview2:MARKer1:Y?` might return 212.3, indicating Marker 1 (M1) is at 212.3 °.

## **CALCulate:TOverview[:MEASview<y>]:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker time for the selected marker on the time overview trace.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands



## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:DELta:X?
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:DELta:Y?](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Returns

<NRf> Delta marker time for the selected marker.

## Examples

CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:DELTA:X? might return 38.0E-9, indicating that the delta marker time is 38.0 ns.

## CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:DELta:Y? (Query Only)

Returns the delta marker amplitude for the selected marker on the time overview trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:DELta:Y?
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:DELta:X?](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Returns

<NRf> Delta marker amplitude for the selected marker.

## Examples

`CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:DELTA:Y?` might return 23.45, indicating that the delta marker amplitude is 23.45 dB on the trace of Time overview Measurement View 1.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the time overview trace.

### Conditions

Measurement views: Time overview displays with specified Measurement view number

### Group

Calculate commands

### Syntax

`CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:MAXimum`

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Examples

`CALCULATE:TOVERVIEW:MEASview1:MARKer1:MARKER1:MAXIMUM` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the time overview trace.

### Conditions

Measurement views: Time overview with specified Measurement view number

### Group

Calculate commands

### Syntax

`CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:HIGHer`

### Related Commands

[\*CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:PEAK:LOWer\*](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:PEAK:HIGHER` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next peak to the left on the time overview trace.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:LEFT
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:PEAK:RIGHT](#)

## Arguments

None

## Examples

`CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the time overview trace.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:LOWer
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:PEAK:HIGHer](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE : TOVERVIEW : MEASVIEW1 : MARKER1 : PEAK : LOWER` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next peak to the right on the time overview trace.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands

## Syntax

`CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:PEAK:RIGHT`

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:PEAK:LEFT](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Examples

`CALCULATE : TOVERVIEW : MEASVIEW1 : MARKER1 : PEAK : RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:X**

Sets or queries the horizontal position of the selected marker on the time overview trace.

## Conditions

Measurement views: Time overview with psecified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:X <value>
```

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:X?
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:Y?](#)

## Arguments

<value> ::= <NRf> specifies the horizontal position of the specified marker.

Range: (analysis offset) to [(analysis offset) + (analysis length)].

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Returns

<NRf> is a floating point number that is the horizontal position of the specified marker.

## Examples

```
CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:X 1.5us places Marker 1 (M1) at 1.5 µs on the trace.
```

## CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the selected marker in the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Calculate commands

## Syntax

```
CALCulate:TOVerview[:MEASview<y>]:MARKer<x>:Y?
```

## Related Commands

[CALCulate:TOVerview\[:MEASview<y>\]:MARKer<x>:X](#)

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

### Returns

<NRf> Marker amplitude of the selected marker.

### Examples

CALCULATE:TOVERVIEW:MEASVIEW1:MARKER1:Y? might return -34.28, indicating Marker 1 (M1) is at -34.28 dBm.

## CALculate:TXGain:MARKer<x>:DELTA:X? (Query Only)

Returns the delta marker frequency for the specified marker on the trace.

### Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

CALCulate:TXGain:MARKer<x>:DELTA:X?

### Returns

<NRf> is the delta marker frequency for the specified marker.

### Examples

CALCulate:TXGain:MARKer1:DELTA:X? might return 1.28E+6, indicating that the delta marker frequency is 1.28 MHz.

## CALculate:TXGain:MARKer<x>:DELTA:Y? (Query Only)

Returns the delta marker amplitude for the specified marker on the trace.

### Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

### Group

Calculate commands

### Syntax

CALCulate:TXGain:MARKer<x>:DELTA:Y?

## Returns

<NRf> is the delta marker amplitude for the specified marker.

## Examples

`CALCulate:TXGain:MARKer1:DELTA:Y?` might return 23.45, indicating that the delta marker amplitude for Marker 1 (M1) is 23.45 dB.

## CALCulate:TXGain:MARKer<x>:MAXimum (No Query Form)

Moves the specified marker to the highest peak on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALCulate:TXGain:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## CALCulate:TXGain:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the specified marker to the next peak higher in amplitude on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:PEAK:HIGHer
```

## Arguments

None

## Examples

`CALCulate:TXGain:MARKer1:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:TXGain:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the specified marker to the next peak to the left on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALCulate:TXGain:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak to the left on the trace.

## **CALCulate:TXGain:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the specified marker to the next peak lower in amplitude on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:PEAK:LOWer
```



## Arguments

None

## Examples

`CALCulate:TXGain:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:TXGain:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the specified marker to the next peak to the right on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALCulate:TXGain:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak to the right on the trace.

## **CALCulate:TXGain:MARKer<x>:TRACe**

Sets or queries the trace on which the specified marker is placed in the Transmission Gain display.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:TRACe { TRACE1 | TRACE2 | TRACE3 }
```

```
CALCulate:TXGain:MARKer<x>:TRACe?
```

## Arguments

TRACE 1 places the specified marker on Trace 1.

TRACE 2 places the specified marker on Trace 2.

TRACE 3 places the specified marker on Trace 3.

## Returns

See Arguments.

## Examples

CALCulate:TXGain:MARKer1:TRACe TRACE1 places Marker 1 (M1) on Trace 1.

## CALCulate:TXGain:MARKer<x>:X

Sets or queries the horizontal position for the specified marker on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Group

Calculate commands

## Syntax

CALCulate:TXGain:MARKer<x>:X <value>

CALCulate:TXGain:MARKer<x>:X ?

## Arguments

<value>::=<NRf> specifies the horizontal position of the specified marker.

Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Returns

<NRf> is the horizontal position of the specified marker.

## Examples

CALCulate:TXGain:MARKer1:X 800E+6 places Marker 1 (M1) at 800 MHz on the trace.

## CALCulate:TXGain:MARKer<x>:Y? (Query Only)

Queries the marker amplitude of the specified marker on the trace.

## Conditions

Measurement view: Transmission Gain

The parameter <x> = 1 to 3; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command.

## Group

Calculate commands

## Syntax

```
CALCulate:TXGain:MARKer<x>:Y?
```

## Returns

<NRf> is the amplitude at the specified marker position in dB.

## Examples

CALCulate:TXGain:MARKer1:Y might return -23.45, indicating that the marker amplitude is -23.45 dB.

## CALCulate:WLAN:AD:CONSte:MARKer<x>:DELTA:X[:TIME]? (Query Only)

Returns the time of the specified delta marker on the trace in the Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AD:CONSte:MARKer<x>:DELTA:X[:TIME]?
```

## Arguments

<x> = 1, 2, 3, 4 is the marker

## Returns

<NR3>

## Examples

CALCulate:WLAN:AD:CONSte:MARKer1:DELTA:X:TIME? might return 3.1494-6, indicating that the frequency of the specified delta between Marker (M1) and MR is 3.1494-6.

## CALCulate:WLAN:AD:CONSte:MARKer<x>:MAGNitude? (Query Only)

Queries the marker magnitude of the specified marker in the Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:CONSte:MARKer<x>:MAGNitude?`

## Arguments

<x> = 1, 2, 3, 4 is the marker

## Returns

<NRf> The magnitude for the selected marker

## Examples

`CALCulate:WLAN:AD:CONSte:MARKer1:MAGNitude?` might return `6.261E-3`, indicating the magnitude of Marker 1 (M1) is `6.261E-3`.

## **CALCulate:WLAN:AD:CONSte:MARKer<x>:PHASe? (Query Only)**

Queries the phase readout of the specified marker in the Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:CONSte:MARKer<x>:PHASe?`

## Arguments

<x> = 1, 2, 3, 4 is the marker

## Returns

<NRf> The phase readout for the selected marker

## Examples

`CALCulate:WLAN:AD:CONSte:MARKer1:PHASe?` might return `9.690`, indicating the phase readout of Marker 1 (M1) is `9.7°`.

## **CALCulate:WLAN:AD:CONSte:MARKer<x>:SYMBol? (Query Only)**

Queries the symbol of the specified marker in Constellation display.

### **Conditions**

Measurement view: 802.11ad Constellation

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:AD:CONSte:MARKer<x>:SYMBol?
```

### **Arguments**

<x> = 1, 2, 3, 4 is the marker

### **Returns**

<NR3> = the symbol of the selected marker

### **Examples**

CALCulate:WLAN:AD:CONSte:MARKer1:SYMBol? might return 3.369E+3, indicating the symbol value of the Marker 1 (M1) is 3.369E+3.

## **CALCulate:WLAN:AD:CONSte:MARKer<x>:TYPE? (Query Only)**

Queries the data type of the specified marker in Constellation display.

### **Conditions**

Measurement view: 802.11ad Constellation

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:AD:CONSte:MARKer<x>:TYPE?
```

### **Arguments**

<x> = 1, 2, 3, 4 is the marker

### **Returns**

<String>

Either (Preamble | Header | Data | Guard)

## Examples

`CALCulate:WLAN:AD:CONStE:MARKer1:TYPE?` might return "Preamble", indicating the data type of the selected marker is "Preamble".

## **CALCulate:WLAN:AD:CONStE:MARKer<x>:VALue? (Query Only)**

Queries the value of the selected marker in Constellation display.

### Conditions

Measurement view: 802.11ad Constellation

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:AD:CONStE:MARKer<x>:VALue?`

### Arguments

<x> = 1, 2, 3, 4 is the marker

### Returns

<NR1> = the value of the selected marker

## Examples

`CALCulate:WLAN:AD:CONStE:MARKer1:VALue?` might return "+1", indicating the value of the Marker 1 (M1) is "+1".

## **CALCulate:WLAN:AD:CONStE:MARKer<x>:X**

Sets or queries the horizontal position of the specified marker on the trace in the 802.11ad Constellation display.

### Conditions

Measurement view: 802.11ad Constellation

<x> = 1, 2, 3, 4 is the marker

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:AD:CONStE:MARKer<x>:X <value> CALCulate:WLAN:AD:CONStE:MARKer<x>:X?`

### Arguments

<value>=<NRf> specifies the horizontal position of the marker

## Returns

<NRf>

## Examples

`CALCulate:WLAN:AD:CONSte:MARKer1:X 9.5409090173E-6` places Marker 1 (M1) at 9.540E-6 on the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:DELTA:X? (Query Only)**

Returns the delta marker time for the selected marker on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:EVM:MARKer<x>:DELTA:X?`

## Arguments

<x>=1,2,3,4 is the marker

## Returns

<NRf> Delta marker time

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:DELTA:X?` might return 1.28E-9, indicating that the delta marker time for Marker 1 (M1) is 1.28E-9 s.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:DELTA:Y? (Query Only)**

Returns the delta marker amplitude for the selected marker on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:EVM:MARKer<x>:DELTA:Y?`

## Arguments

<x>=1,2,3,4 is the marker

## Returns

<NRf> = Delta marker amplitude

## Examples

`CALCulate:WLAN:AD:EVM:MARKer<x>:DELTA:Y?` might return 23.45, indicating that the delta marker amplitude for Marker 1 (M1) is 23.45 dB.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:MAXimum (No Query Form)**

Moves the selected marker to the highest peak on the trace in the EVM measurement.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:EVM:MARKer<x>:MAXimum`

## Arguments

<x>=1,2,3,4 is the marker

## Returns

None

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:MAXimum` moves Marker 1 (M1) to the highest peak on the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the selected marker to the next peak higher in amplitude on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands



## Syntax

```
CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:HIGHer
```

## Arguments

<x>=1,2,3,4 is the marker

## Returns

None

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:PEAK:HIGHer` moves Marker 1 (M1) to the next peak higher in amplitude on the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the selected marker to the next significant peak found to the left of the current marker position.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:LEFT
```

## Arguments

<x>=1,2,3,4 is the marker

## Returns

None

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:PEAK:LEFT` moves Marker 1 (M1) to the next peak found on the left side of the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the selected marker to the next peak lower in amplitude on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:LOWer`

## Arguments

<x>=1,2,3,4 is the marker

## Returns

None

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:PEAK:LOWer` moves Marker 1 (M1) to the next peak lower in amplitude on the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the selected marker to the next significant peak found to the right of the current marker position.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:RIGHT`

## Arguments

<x>=1,2,3,4 is the marker

## Returns

None

## Examples

`CALCulate:WLAN:AD:EVM:MARKer1:PEAK:RIGHT` moves Marker 1 (M1) to the next peak found on the right side of the trace.

## **CALCulate:WLAN:AD:EVM:MARKer<x>:X**

Sets or queries the horizontal (time) stamp of the selected marker on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AD:EVM:MARKer<x>:X <value> CALCulate:WLAN:AD:EVM:MARKer<x>:X?
```

## Arguments

where

<x>=1,2,3,4 is the marker

<value>= <NRf>

## Returns

<NRf>

## Examples

CALCulate:WLAN:AD:EVM:MARKer<x>:PEAK:LOWer 5.36 places Marker 1 (M1) at 5.36  $\mu$ s on the EVM trace.

## CALCulate:WLAN:AD:EVM:MARKer<x>:Y? (Query Only)

Queries the vertical position of the selected marker on the EVM trace.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AD:EVM:MARKer<x>:Y?
```

## Arguments

<x>=1,2,3,4 is the marker

## Returns

<NRf> Vertical position of marker

## Examples

CALCulate:WLAN:AD:EVM:MARKer1:Y? might return -34.28, indicating the vertical position of Marker 1 (M1) is -34.28 dBm on the EVM trace.

## CALCulate:WLAN:AY:CONSte:MARKer<x>:DELTA:X:[ :TIME]? (Query Only)

Query the value of the specified delta marker on the trace in the Constellation display.

You must activate the specified marker by using CALCulate:MARKer:ADD.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:DELTA:X[:TIME]?
```

## Returns

<NRf>

Units: s (seconds)

## Examples

`CALC:WLAN:AY:CONS:MARK1:DELT:X:TIME?` may return `-1.772e-6`, indicating that `-1.772e-6` seconds is value of the specified delta marker between Marker 1 and the Reference Marker.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:MAGNitude? (Query Only)

Query the marker magnitude of the specified marker in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:MAGNitude?
```

## Returns

<NRf> is the magnitude for the specified delta marker.

## Examples

`CALC:WLAN:AY:CONS:MARK2:MAGN?` may return `6.261e-3`, indicating that the magnitude for Marker 2 is `6.261e-3`.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:PHASe? (Query Only)

Query the phase readout of the specified marker in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:PHASe?
```

## Returns

<NRf> is the phase readout for the specified marker. Units: degrees

## Examples

CALC:WLAN:AY:CONS:MARK3:PHAS? may return 9.690, indicating that the phase readout for Marker 3 is 9.7° degrees.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:SYMBol? (Query Only)

Query the Symbol value at which the specified marker is placed.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands.

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:SYMBol?
```

## Returns

<NR1>

## Examples

CALC:WLAN:AY:CONS:MARK1:SYMB? may return 3.369e+3, indicating that the symbol value at which Marker 1 is placed is 3.369e+3.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:TYPE? (Query Only)

Query the data type of the specified marker in the Constellation display.

The data type can be Preamble, Header, Data, or Guard, based on the marker position selection in the region.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:TYPE?
```

## Returns

<string>

## Examples

CALC:WLAN:AY:CONS:MARK1:TYPE? may return Preamble, indicating that the data type for Marker 1 is Preamble.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:VALue? (Query Only)

Query the symbol value at which the marker is placed.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:VALue?
```

## Returns

<string>

## Examples

CALC:WLAN:AY:CONS:MARK1:VAL? may return +1, which indicates the symbol value at which Marker 1 is placed.

## CALCulate:WLAN:AY:CONStE:MARKer<x>:X

Set or query the horizontal position of the specified marker on the trace in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

<x> = 1 to 4 (0 is the reference marker; it is invalid).

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:X <value>
```

```
CALCulate:WLAN:AY:CONStE:MARKer<x>:X?
```

## Arguments

<value> : :=<NRf> specifies the horizontal position of the specified marker.

Range: 0.1 to 10 s (seconds) s

## Returns

<NRf>

## Examples

CALC:WLAN:AY:CONS:MARK1:X 9.5409090173e-6 places the horizontal position of Marker 1 at 9.540e-6 on the trace.

CALC:WLAN:AY:CONS:MARK2:X? may return 9.540e-6, which is the horizontal position of Marker 2 on the trace.

## CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:X

Set or query the delta marker value for the specified marker on the EVM trace.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker, is invalid)

You must activate the specified marker by using CALCulate:MARKer:ADD.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:X <NRf>
```

```
CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:X?
```

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

<NRf> is the delta marker value for the specified marker.

Units: s (seconds)

## Returns

<NRf>

## Examples

`CALC:WLAN:AY:EVM:MARK1:DELTA:X 1.28e-9` sets the delta marker value for Marker 1 to 1.28e-9 s.

`CALC:WLAN:AY:EVM:MARK2:DELTA:X?` may return 1.28e-9, indicating that the delta marker value for Marker 2 is 1.28e-9 s.

## **CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:Y**

Set or query the delta marker amplitude for the specified marker on the EVM trace.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker, is invalid)

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:Y <NRf>`

`CALCulate:WLAN:AY:EVM:MARKer<x>:DELTA:Y?`

## Related Commands

[\*CALCulate:MARKer:ADD\*](#)

## Arguments

<NRf> is the delta marker amplitude for the specified marker.

Units: dB

## Returns

<NRf>

## Examples

`CALC:WLAN:AY:EVM:MARK1:DELTA:Y 23.45` sets the delta marker amplitude of Marker 1 to 23.45 dB.

`CALC:WLAN:AY:EVM:MARK2:DELTA:Y?` may return 23.45, which means the marker amplitude of Marker 2 is 23.45 dB.

## **CALCulate:WLAN:AY:EVM:MARKer<x>:MAXimum (No Query Form)**

Move the specified marker to the highest peak on the trace in the EVM display.



## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:EVM:MARKer<x>:MAXimum
```

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

None

## Examples

```
CALC:WLAN:AY:EVM:MARK1:MAX
```

 moves Marker 1 to the highest peak on the trace.

## **CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:HIGHer (No Query Form)**

Move the specified marker to the next peak higher in amplitude on the EVM trace.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:HIGHer
```

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

None

## Examples

`CALC:WLAN:AY:EVM:MARK1:PEAK:HIGH` moves Marker 1 to the next peak higher in amplitude on the trace.

## **CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:LEFT (No Query Form)**

Move the specified marker to the next significant peak found to the left of the current marker position.

### Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:LEFT`

### Related Commands

[CALCulate:MARKer:ADD](#)

### Arguments

None

## Examples

`CALC:WLAN:AY:EVM:MARK4:PEAK:LEFT` moves Marker 4 to the next peak to the left on the trace.

## **CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:LOWer (No Query Form)**

Move the specified marker to the next peak lower in amplitude on the EVM trace.

### Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:LOWer`

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

None

## Examples

`CALC:WLAN:AY:EVM:MARK2:PEAK:LOW` moves Marker 2 to the next peak lower in amplitude on the trace.

## CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:RIGHT (No Query Form)

Move the specified marker to the next significant peak found to the right of the current marker position.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:AY:EVM:MARKer<x>:PEAK:RIGHT`

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

None

## Examples

`CALC:WLAN:AY:EVM:MARK1:PEAK:RIGHT` moves Marker 1 to the next peak to the right on the trace.

## CALCulate:WLAN:AY:EVM:MARKer<x>:X

Set or query the horizontal (time) stamp of the specified marker on the EVM trace.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker; it is invalid).

You must activate the specified marker by using `CALCulate:MARKer:ADD`.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:EVM:MARKer<x>:X <value> CALCulate:WLAN:AY:EVM:MARKer<x>:X?
```

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

<value>: :=<NRf> is the specified marker.

Units:  $\mu$ s

## Returns

<NRf>

## Examples

CALC:WLAN:AY:EVM:MARK1:X 5e-6 places Marker 1 at 5  $\mu$ s on the EVM trace.

CALC:WLAN:AY:EVM:MARK2:X? may return 5e-6, which means Marker 2 is placed at 5  $\mu$ s on the EVM trace.

## CALCulate:WLAN:AY:EVM:MARKer<x>:Y

Set or query the vertical (EVM) position of the specified marker on the EVM trace.

## Conditions

Measurement views: 802.11ay Analysis EVM

<x> = 1 to 4 (0 is the reference marker, is invalid)

You must activate the specified marker by using CALCulate:MARKer:ADD.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:AY:EVM:MARKer<x>:Y <NRf> CALCulate:WLAN:AY:EVM:MARKer<x>:Y?
```

## Related Commands

[CALCulate:MARKer:ADD](#)

## Arguments

<NRf> is the marker amplitude of the specified marker.

Units: dB

## Returns

<NRf>

## Examples

`CALC:WLAN:AY:EVM:MARK1:Y -34.28` sets the position of Marker 1 at -34.28 dB on the EVM trace.

`CALC:WLAN:AY:EVM:MARK2:Y?` may return -34.28, indicating that Marker 2 is at -34.28 dB on the EVM trace.

## CALCulate:WLAN:CONSte:MARKer<x>:FREQuency

Sets or queries the frequency value of the marker on the WLAN Constellation display. The units are either Frequency in Hz or Subcarrier number. To set the units, use the command `[SENSe]:WLAN:UNIT:FREQuency`.

Note that FREQuency is not an actual measurement of frequency, but is the subcarrier offset within the analyzed packet burst.

<x>, which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CONSte:MARKer<x>:FREQuency <NRf>`

`CALCulate:WLAN:CONSte:MARKer<x>:FREQuency?`

## Arguments

Floating point number that indicates the desired subcarrier to which the marker should be moved. Subcarriers are traditionally numbered with both positive and negative offsets from the CF for the modulation scheme. Valid integral values range from `-maxsubcarrier` to `maxsubcarrier`, where the actual values depend on the modulation standard. Specifying a value of <NRf> outside of the range of subcarriers present in the burst selects the nearest valid subcarrier. Subcarriers can also be specified by frequency (also determined by modulation standard).

## Returns

Returns a unitless number representing the subcarrier number.

## Examples

`CALC:WLAN:CONST:MARK2:FREQ 2.0000` sets the position of Marker 2 (M2) to Subcarrier 2 (assuming the units have been set to Subcarrier).

`CALC:WLAN:CONST:MARK2:FREQ?` might return 2.41575E+9, indicating that Marker 2 (M2) is positioned at 2.41575 MHz.

## **CALCulate:WLAN:CONSte:MARKer<x>:MAGNitude? (Query Only)**

Returns the magnitude value at the indicated marker position on the WLAN Constellation display.

<x> is optional. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:CONSte:MARKer<x>:MAGNitude?
```

### **Returns**

Returns a unitless, floating point number indicating the magnitude at the selected marker position on the WLAN Constellation display.

### **Examples**

```
CALC:WLAN:CONS:MARK1:MAGN? might return 35 indicating the magnitude at Marker 1 (M1) is 35.
```

## **CALCulate:WLAN:CONSte:MARKer<x>:PHASe? (Query Only)**

Returns the phase value at the indicated marker position on the WLAN Constellation display, in degrees.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:CONSte:MARKer<x>:PHASe?
```

### **Returns**

Phase value of the signal in degrees between -180 and +180 as an ASCII encoded floating point number in standard engineering notation.

## Examples

`CALC:WLAN:CONS:MARK1:PHAS?` might return 35.74, indicating the phase readout of Marker 1 (M1) is 35.74°.

## CALCulate:WLAN:CONSte:MARKer<x>:TIME

Sets the time offset value at the selected marker position on the points trace on the WLAN Constellation display. The units are either Symbols or Seconds. To set/query the units, use the command `[SENSe]:WLAN:UNIT:TIME`.

Note that TIME is a measurement of neither absolute nor elapsed time, but symbol offset within the analyzed packet burst.

<x>, which is optional, is the desired marker. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CONSte:MARKer<x>:TIME <Nrf> CALCulate:WLAN:CONSte:MARKer<x>:TIME?`

## Arguments

Floating point number that indicates the 0 based symbol or time offset from the start of packet burst. Valid integral values range from 0 to the number of symbols in the burst. Specifying a value outside of the range of symbols present in the burst selects the nearest valid symbol.

## Returns

Returns a unitless integer value in floating point format that represents the symbol associated with the position of the indicated marker, or the time offset.

## Examples

`CALC:WLAN:CONST:MARK3:TIME -234.5` moves Marker 3 (M3) to the 362.42µS position on the trace (assuming the units have been set to Seconds).

`CALC:WLAN:CONST:MARK3:TIME?` might return 5.0000 indicating that Marker 3 is located at symbol 5 (assuming units have been set to Symbols).

## CALCulate:WLAN:CONSte:MARKer<x>:TYPE? (Query Only)

Returns the data type of the subcarrier for data at the indicated marker position - either pilot or data.

<x>, which is optional, is the desired marker. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:CONStE:MARKer<x>:TYPE?
```

## Returns

Returns the decoded value of the subcarrier type to which the marker is attached as an ASCII encoded floating point number in standard engineering notation.

Values are:

1.000 if the marker type is a pilot marker.

2.000 if the marker type is a data marker.

## Examples

`CALC:WLAN:CONS:MARK2:TYPE?` might return 1.000 indicating Marker 2 (M2) is a pilot marker.

## CALCulate:WLAN:CONStE:MARKer<x>:VALue? (Query Only)

Queries the value readout at the selected marker position on the points trace of the WLAN Constellation display.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral

values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:CONStE:MARKer<x>:VALue?
```

## Returns

Floating point number that represents the value readout at the selected marker position.



## Examples

`CALC:WLAN:CONS:MARK1:VAL?` might return 2.00, indicating the value readout at Marker 1 (M1) is 2.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:X? (Query Only)**

Queries the delta Subcarrier or frequency value (depending on which units have been set) at the selected marker position on the WLAN Channel Response display.

Use the command `[SENSe]:WLAN:UNIT:FREQuency` to set the units.

<x> is the desired marker to query. Marker 0 is not valid for the DELTA queries. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:X?`

## Returns

Floating point number that represents the delta frequency value at the selected marker position.

## Examples

`CALC:WLAN:CRES:MARK2:DELTA:X?` might return 1.28E+6 indicating that the delta frequency value for Marker 2 (M2) is 1.28 MHz.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:Y? (Query Only)**

Queries the vertical delta value for the points trace at the selected marker position on the WLAN Channel Response display, in dB or degree.

<x> is the desired marker to query. Marker 0 is not valid for the DELTA queries. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

Marker 0 and the selected marker must be on the same plot (magnitude or phase); otherwise, it will result in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:DELTA:Y?`

## Returns

Returns the vertical delta value for the points trace in dB or degree at the selected marker position.

## Examples

`CALC:WLAN:CRES:MARK3:DELT:Y?` might return `-30`, indicating a vertical delta value of -30 dB at the Marker 3 (M3) position.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:MAXimum (No Query Form)**

Moves the marker to the highest peak on the trace on the WLAN Channel Response display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:MAXimum`

## Arguments

None

## Examples

`CALC:WLAN:CRES:MARK2:MAX` moves the marker to the highest peak on the trace on the WLAN Channel Response display.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the marker to the next peak higher in amplitude on the WLAN Channel Response display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:HIGHer`

## Arguments

None

## Examples

`CALC:WLAN:CRES:MARK2:PEAK:HIGHER` moves Marker 2 (M2) to the next peak higher in amplitude on the trace.

## **CALCulate:WLAN:CRESponse:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the marker to the next peak on the left on the trace in the WLAN Channel Response display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:CRESponse:MARKer<x>:PEAK:LEFT
```

## Arguments

None

## Examples

`CALC:WLAN:CRES:MARK2:PEAK:LEFT` moves Marker 2 (M2) to the next peak on the left on the trace in the WLAN Channel Response display.

## **CALCulate:WLAN:CRESponse:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the marker to the next peak lower in amplitude on the trace on the WLAN Channel Response display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:CRESponse:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALC:WLAN:CRES:MARK2:PEAK:LOW` moves Marker 2 (M2) to the next peak lower on the trace.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the marker to the next peak to the right on the trace on the WLAN Channel Response display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:PEAK:RIGHT`

### Arguments

None

## Examples

`CALC:WLAN:CRES:MARK2:PEAK:RIGHT` moves Marker 2 (M2) to the next peak to the next peak on the right on the trace.

## **CALCulate:WLAN:CRESPonse:MARKer<x>:X**

Sets or queries the value of the horizontal position (units in Frequency or Subcarrier) for the selected marker on the points trace in the WLAN Channel Response display.

To set the units, use the command `[SENSe]:WLAN:UNIT:FREQuency`.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:CRESPonse:MARKer<x>:X <NRf>`

`CALCulate:WLAN:CRESPonse:MARKer<x>:X?`

### Arguments

Floating point number that specifies the value of the horizontal position of the marker. Range: Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

`CALC:WLAN:CRES:MARK3:X 22` places Marker 3 at 22 Subcarrier on the trace (assuming the units have been set to Subcarrier).

## CALCulate:WLAN:CRESponse:MARKer<x>:Y? (Query Only)

Returns the value of the amplitude at the selected marker position in the WLAN Channel Response measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:CRESponse:MARKer<x>:Y?`

### Returns

Floating point number that indicates the amplitude value at the selected marker position in dB or degrees.

## Examples

`CALC:WLAN:CRES:MARK1:Y?` might return `33.393856E-3` indicating the amplitude of Marker 1 is 0.033 dB.

## CALCulate:WLAN:EVM:MARKer<x>:AVERage:SUBCarrier? (Query Only)

Queries the EVM value at the selected marker position on the average trace on the WLAN EVM display subcarrier graph.

To query the marker value of the points trace, use the query.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:EVM:MARKer<x>:AVERage:SUBCarrier?`

## Returns

Returns a floating point number that represents the EVM value at the selected marker position on the average trace on the WLAN EVM display subcarrier graph.

## Examples

`CALC:WLAN:EVM:MARK3:AVER:SUBC?` might return 16.4, indicating an EVM value of 16.4 percent at the Marker 3 (M3) position on the average trace on the WLAN EVM display subcarrier graph.

## CALCulate:WLAN:EVM:MARKer<x>:AVERage:SYMBOL? (Query Only)

Queries the EVM value at the selected marker position on the average trace on the WLAN EVM display symbol graph.

To query the marker value of the points trace, use the query.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not

supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:EVM:MARKer<x>:AVERage:SYMBOL?`

## Returns

Returns a floating point number that represents the EVM value at the selected marker position on the average trace on the WLAN EVM display symbol graph.

## Examples

`CALC:WLAN:EVM:MARK3:AVER:SYMB?` might return 14.5, indicating an EVM value of 14.5 percent at the Marker 3 (M3) position on the average trace on the WLAN EVM display symbol graph.

## CALCulate:WLAN:EVM:MARKer<x>:DELTA:SUBCarrier? (Query Only)

Queries the delta Subcarrier or frequency value (depending on which units have been set) at the selected marker position on the WLAN EVM display.

Use the command `[SENSe]:WLAN:UNIT:FREQuency` to set the units.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:EVM:MARKer<x>:DELTA:SUBCarrier?
```

## Returns

Returns a floating point number that represents the delta subcarrier or frequency value at the selected marker position on the WLAN EVM display.

## Examples

`CALC:WLAN:EVM:MARK3:DELTA:SUBCarrier?` might return 9.375 MHz, indicating a delta frequency value of 9.375 MHz at the Marker 3 (M3) position.

## CALCulate:WLAN:EVM:MARKer<x>:DELTA:SYMBOL? (Query Only)

Queries the delta time value in seconds or symbols (depending on which units have been set) at the selected marker position on the WLAN EVM display.

Use the command `[SENSe]:WLAN:UNIT:TIME` to set the units.

`<x>`, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If `<x>` is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:EVM:MARKer<x>:DELTA:SYMBOL?
```

## Returns

Returns the delta time value at the selected marker position on the WLAN EVM display.

## Examples

`CALC:WLAN:EVM:MARK3:DELTA:SYMB?` might return 4.0E-6  $\mu$ s, indicating a delta time value of 4  $\mu$ s at the Marker 3 (M3) position (assuming the units have been set to Seconds).

## **CALCulate:WLAN:EVM:MARKer<x>:DELTA:Y? (Query Only)**

Queries the vertical delta value for the points trace at the selected marker position on the WLAN EVM display.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral

values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:EVM:MARKer<x>:DELTA:Y?
```

### **Returns**

Returns the vertical delta value for the points trace at the selected marker position.

### **Examples**

`CALC:WLAN:EVM:MARK3:DELTA:Y?` might return `-212.91`, indicating a vertical delta value of -212.91 percent at the Marker 3 (M3) position.

## **CALCulate:WLAN:EVM:MARKer<x>:FREQUENCY**

Sets or queries the frequency value of the marker on the WLAN EVM display. The units are either Frequency in Hz or Subcarrier number. To set the units, use the command `[SENSe]:WLAN:UNIT:FREQUENCY`.

Note that FREQUENCY is not an actual measurement of frequency, but is the subcarrier offset within the analyzed packet burst.

<x>, which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:EVM:MARKer<x>:FREQUENCY <NRf>
```

```
CALCulate:WLAN:EVM:MARKer<x>:FREQUENCY?
```



## Arguments

Floating point number that indicates the desired subcarrier or frequency to which the marker should be moved. Subcarriers are traditionally numbered with both positive and negative offsets from the CF for the modulation scheme. Valid integral values range from  $-\text{maxsubcarrier}$  to  $\text{maxsubcarrier}$ , where the actual values depend on the modulation standard. Specifying a value of  $\langle \text{NRf} \rangle$  outside of the range of subcarriers present in the burst selects the nearest valid subcarrier. Subcarriers can also be specified by frequency (also determined by modulation standard).

## Returns

Returns a unitless number representing the subcarrier number or frequency in Hz.

## Examples

`CALC:WLAN:EVM:MARK2:FREQ 2.0000` sets the position of Marker 2 (M2) to Subcarrier 2 (assuming the units have been set to Subcarrier).

`CALC:WLAN:EVM:MARK2:FREQ?` might return `2.41575E+9`, indicating that Marker 2 (M2) is positioned at 2.41575 MHz.

## CALCulate:WLAN:EVM:MARKer<x>:TIME

Sets the vertical marker value of the points trace on the WLAN EVM display. The units are either Symbols or Seconds. To set/query the units, use the command `[SENSe]:WLAN:UNIT:TIME`.

Note that TIME is a measurement of neither absolute nor elapsed time, but symbol offset within the analyzed packet burst.

$\langle x \rangle$ , which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If  $\langle x \rangle$  is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:EVM:MARKer<x>:TIME <NRf> CALCulate:WLAN:EVM:MARKer<x>:TIME?`

## Arguments

Floating point number that indicates the 0 based symbol or time offset from the start of packet burst. Valid integral values range from 0 to the number of symbols

in the burst. Specifying a value outside of the range of symbols present in the burst selects the nearest valid symbol.

## Returns

Returns a unitless integer value in floating point format that represents the symbol associated with the position of the indicated marker, or the time offset.

## Examples

`CALC:WLAN:EVM:MARK3:TIME 362.42E-6` moves Marker 3 (M3) to the 362.42µs position on the trace (assuming the units have been set to Seconds).

`CALC:WLAN:EVM:MARK3:TIME?` might return 5.0000 indicating that Marker 3 is located at symbol 5 (assuming units have been set to Symbols).

## CALCulate:WLAN:EVM:MARKer<x>:VALue? (Query Only)

Queries the EVM value at the selected marker position on the points trace of the WLAN EVM display.

The EVM value of a marker on the average trace is queried by using the and `CALCulate:WLAN:EVM:MARKer<x>:AVERAge:SYMBol?` queries.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:EVM:MARKer<x>:VALue?`

## Returns

Returns the EVM value at the selected marker position on the WLAN EVM display.

## Examples

`CALC:WLAN:EVM:MARK3:VAL?` might return 11.9, indicating an EVM value of 11.9 percent for Marker 3 (M3).

## CALCulate:WLAN:FLATness:MARKer<x>:MAXimum (No Query Form)

Positions the specified marker at the maximum point on the trace in the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:FLATness:MARKer<x>:MAXimum
```

## Arguments

None

## Examples

`CALC:WLAN:FLAT:MARK2:MAX` positions the Marker 2 (M2) at the maximum point on the trace.

## CALCulate:WLAN:FLATness:MARKer<x>:PEAK:HIGHer (No Query Form)

Moves the specified marker to the next peak higher in amplitude on the trace on the WLAN Spectral Flatness display.



**Note:** *WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.*

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:HIGHer
```

## Arguments

None

## Examples

`CALC:WLAN:FLAT:MARK2:PEAK:HIGH` moves Marker 2 (M2) to the next peak higher in amplitude on the trace.

## CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LEFT (No Query Form)

Moves the specified marker to the next peak to the left on the trace in the WLAN Spectral Flatness display.



**Note:** *WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.*

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LEFT`

## Arguments

None

## Examples

`CALC:WLAN:FLAT:MARK2:PEAK:LEFT` moves Marker 2 (M2) to the next peak to the left on the trace .

## **CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the marker to the next peak lower in amplitude on the trace on the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:FLATness:MARKer<x>:PEAK:LOWer`

## Arguments

None

## Examples

`CALC:WLAN:FLAT:MARK2:PEAK:LOW` moves Marker 2 (M2) to the next peak lower on the trace.

## **CALCulate:WLAN:FLATness:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the marker to the next peak to the right on the trace on the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:FLATness:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

```
CALC:WLAN:FLAT:MARK2:PEAK:RIGH moves Marker 2 (M2) to the next peak to the next peak on the right on the trace.
```

## CALCulate:WLAN:FLATness:MARKer<x>:X

Sets or queries the horizontal position (units in Frequency or Subcarrier) for the selected marker on the points trace in the WLAN Spectral Flatness display.

To set the units, use the command [SENSe]:WLAN:UNIT:FREQuency.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:FLATness:MARKer<x>:X <NRf> CALCulate:WLAN:FLATness:MARKer<x>:X?
```

## Arguments

Floating point number that specifies the horizontal position of the marker. Range:

Start to Stop frequency (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

```
CALC:WLAN:FLAT:MARK3:X 22 places Marker 3 at 22 Subcarrier on the trace (assuming the units have been set to Subcarrier).
```

## CALCulate:WLAN:FLATness:MARKer<x>:Y? (Query Only)

Returns the value of the amplitude (vertical position) at the selected marker position in the WLAN Spectral Flatness measurement.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:FLATness:MARKer<x>:Y?
```

## Returns

Floating point number that indicates the amplitude at the selected marker position in dB.

## Examples

`CALC:WLAN:FLAT:MARK1:Y?` might return `33.393856E-3` indicating the amplitude of Marker 1 is 0.033 dB.

## **CALCulate:WLAN:MERRor:MARKer<x>:AVERage:SUBCarrier? (Query Only)**

Queries the value readout at the selected marker position on the average trace on the WLAN Magnitude Error display.

To query the value readout on the symbol graph, use `CALCulate:WLAN: MERRor:MARKer<x>:AVERage:SYMBOL?`

`<x>`, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If `<x>` is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:AVERage:SUBCarrier?
```

## Returns

Returns a floating point number that represents the value readout at the selected marker position on the average trace on the WLAN Magnitude Error display subcarrier graph.

## Examples

`CALC:WLAN:MERR:MARK2:AVERAGE:SUBC?` might return `16.4`, indicating a value readout of 16.4 percent at the Marker 3 (M3) position on the average trace on the WLAN Magnitude Error display subcarrier graph.

## **CALCulate:WLAN:MERRor:MARKer<x>:AVERage:SYMBOL? (Query Only)**

Queries the value readout at the selected marker position on the average trace on the WLAN Magnitude Error display.

To query the value readout on the subcarrier graph, use `CALCulate:WLAN: MERRor:MARKer<x>:AVERage:SUBCarrier?`

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:AVERage:SYMBOL?
```

## Returns

Returns a floating point number that represents the value readout at the selected marker position on the average trace on the WLAN Magnitude Error display symbol graph.

## Examples

`CALC:WLAN:MERR:MARK2:AVERAGE:SUBC?` might return 14.5, indicating a Magnitude Error value of 14.5 percent at the Marker 3 (M3) position on the average trace on the WLAN Magnitude Error display.

## CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SUBCarrier? (Query Only)

Queries the delta Subcarrier or frequency value (depending on which units have been set) at the selected marker position on the WLAN Magnitude Error display.

Use the command `[SENSe]:WLAN:UNIT:FREQuency` to set the units.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SUBCarrier?
```

## Returns

Returns a floating point number that represents the delta subcarrier or frequency value at the selected marker position on the WLAN Magnitude Error display.

## Examples

`CAL:WLAN:MERR:MARK2:DELTA:SUBC?` might return 9.375 MHz, indicating a delta frequency value of 9.375 MHz at the Marker 3 (M3) position.

## **CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SYMBOL? (Query Only)**

Queries the delta time value in seconds or symbols (depending on which units have been set) at the selected marker position on the WLAN Magnitude Error display.

Use the command `[SENSe]:WLAN:UNIT:TIME` to set the units.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

`CALCulate:WLAN:MERRor:MARKer<x>:DELTA:SYMBOL?`

## Returns

Returns the delta time value at the selected marker position on the WLAN Magnitude Error display.

## Examples

`CALC:WLAN:MERR:MARK3:DELTA:SYMB?` might return 4.0E-6  $\mu$ s, indicating a delta time value of 4  $\mu$ s at the Marker 3 (M3) position (assuming the units have been set to Seconds).

## **CALCulate:WLAN:MERRor:MARKer<x>:DELTA:Y? (Query Only)**

Queries the vertical delta value for the points trace at the selected marker position on the WLAN Magnitude Error display.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands



## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:DELta:Y?
```

## Returns

Returns the vertical delta value for the points trace at the selected marker position.

## Examples

`CALC:WLAN:MERR:MARK3:DELta:Y?` might return `-212.91`, indicating a vertical delta value of -212.91 percent at the Marker 3 (M3) position.

## CALCulate:WLAN:MERRor:MARKer<x>:FREQuency

Sets or queries the frequency value of the marker on the WLAN Magnitude Error display. The units are either Frequency in Hz or Subcarrier number. To set the units, use the command `[SENSe]:WLAN:UNIT:FREQuency`.

Note that `FREQuency` is not an actual measurement of frequency, but is the subcarrier offset within the analyzed packet burst.

`<x>`, which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If `<x>` is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:FREQuency <NRf>
CALCulate:WLAN:MERRor:MARKer<x>:FREQuency?
```

## Arguments

Floating point number that indicates the desired subcarrier or frequency to which the marker should be moved. Subcarriers are traditionally numbered with both positive and negative offsets from the CF for the modulation scheme. Valid integral values range from `-maxsubcarrier` to `maxsubcarrier`, where the actual values depend on the modulation standard. Specifying a value of `<NRf>` outside of the range of subcarriers present in the burst selects the nearest valid subcarrier. Subcarriers can also be specified by frequency (also determined by modulation standard).

## Returns

Returns a unitless number representing the subcarrier number or frequency value in Hz.

## Examples

`CALC:WLAN:MERR:MARK2:FREQ 2.0000` sets the position of Marker 2 (M2) to Subcarrier 2 (assuming the units have been set to Subcarrier).

`CALC:WLAN:MERR:MARK2:FREQ?` might return `2.41575E+9`, indicating that Marker 2 (M2) is positioned at 2.41575 MHz.

## CALCulate:WLAN:MERRor:MARKer<x>:TIME

Sets the time offset value at the selected marker position on the points trace on the WLAN Magnitude Error display. The units are either Symbols or Seconds. To set/query the units, use the command [SENSe]:WLAN:UNIT:TIME.

Note that TIME is a measurement of neither absolute nor elapsed time, but symbol offset within the analyzed packet burst.

<x>, which is optional, is the desired marker. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:TIME <Nrf> CALCulate:WLAN:MERRor:MARKer<x>:TIME?
```

### Arguments

Floating point number that indicates the 0 based symbol or time offset from the start of packet burst. Valid integral values range from 0 to the number of symbols in the burst. Specifying a value outside of the range of symbols present in the burst selects the nearest valid symbol.

### Returns

Returns a unitless integer value in floating point format that represents the time associated with the position of the indicated marker, or the time offset.

### Examples

`CALC:WLAN:MERR:MARK3:TIME 362.42E-6` moves Marker 3 (M3) to the 362.42  $\mu$ S position on the trace (assuming the units have been set to Seconds).

`CALC:WLAN:MERR:MARK3:TIME?` might return 5.0000 indicating that Marker 3 is located at symbol 5 (assuming units have been set to Symbols).

## CALCulate:WLAN:MERRor:MARKer<x>:VALue? (Query Only)

Queries the value readout at the selected marker position on the points trace of the WLAN Magnitude Error display.

The value readout of a marker on the average trace is queried by using the and `CALCulate:WLAN:MERRor:MARKer<x>:AVERage:SYMBol?` queries.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:MERRor:MARKer<x>:VALue?
```

## Returns

Returns the decoded symbol value at the selected marker position on the WLAN Magnitude Error display.

## Examples

`CALC:WLAN:MERR:MARK3:VAL?` might return 11.9, indicating a value readout of 11.9 for Marker 3 (M3).

## **CALCulate:WLAN:PERRor:MARKer<x>:AVERage:SUBCarrier? (Query Only)**

Queries the Phase Error value at the selected marker position on the average trace on the WLAN Phase Error display subcarrier graph.

To query the marker value of the points trace, use the query.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:AVERage:SUBCarrier?
```

## Returns

Returns a floating point number that represents the Phase Error value at the selected marker position on the average trace on the WLAN Phase Error display subcarrier graph.

## Examples

`CALC:WLAN:PERR:MARK2:AVERAGE:SUBC?` might return 16.4, indicating a Phase Error value of 16.4 percent at the Marker 3 (M3) position on the average trace on the WLAN Phase Error display subcarrier graph.

## **CALCulate:WLAN:PERRor:MARKer<x>:AVERage:SYMBOL? (Query Only)**

Queries the Phase Error value at the selected marker position on the average trace on the WLAN Phase Error display symbol graph.

To query the marker value of the points trace, use the query.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:AVERage:SYMBOL?
```

## Returns

Returns a floating point number that represents the Phase Error value at the selected marker position on the average trace on the WLAN Phase Error display symbol graph.

## Examples

`CALC:WLAN:PERR:MARK2:AVERAGE:SUBC?` might return 14.5, indicating a Phase Error value of 14.5 percent at the Marker 3 (M3) position on the average trace on the WLAN Phase Error display symbol graph.

## **CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SUBCarrier? (Query Only)**

Queries the delta Subcarrier or frequency value (depending on which units have been set) at the selected marker position on the WLAN Phase Error display.

Use the command `[SENSe]:WLAN:UNIT:FREQuency` to set the units.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SUBCarrier?
```

## Returns

Returns a floating point number that represents the delta subcarrier or frequency value at the selected marker position on the WLAN Phase Error display.

## Examples

`CAL:WLAN:PERR:MARK2:DELTA:SUBC?` might return 9.375 MHz, indicating a delta frequency value of 9.375 MHz at the Marker 3 (M3) position.

## CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SYMBOL? (Query Only)

Queries the delta time value in seconds or symbols (depending on which units have been set) at the selected marker position on the WLAN Phase Error display.

Use the command [SENSe]:WLAN:UNIT:TIME to set the units.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:SYMBOL?
```

### Returns

Returns the delta time value at the selected marker position on the WLAN Phase Error display.

### Examples

CALC:WLAN:PERR:MARK3:DELTA:SYMB? might return 4.0E-6  $\mu$ s, indicating a delta time value of 4  $\mu$ s at the Marker 3 (M3) position (assuming the units have been set to Seconds).

## CALCulate:WLAN:PERRor:MARKer<x>:DELTA:Y? (Query Only)

Queries the vertical delta value for the points trace at the selected marker position on the WLAN Phase Error display.

<x>, which is optional, is the desired marker to query. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Marker 0 is not valid for the DELTA queries. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:DELTA:Y?
```

## Returns

Returns the vertical delta value for the points trace at the selected marker position.

## Examples

`CALC:WLAN:PERR:MARK3:DELTA:Y?` might return `-212.91`, indicating a vertical delta value of -212.91 percent at the Marker 3 (M3) position.

## CALCulate:WLAN:PERRor:MARKer<x>:FREQUency

Sets or queries the frequency value at the selected marker position on the WLAN Phase Error display. The units are either Frequency in Hz or Subcarrier number. To set the units, use the command `[SENSE]:WLAN:UNIT:FREQUency`.

Note that `FREQUency` is not an actual measurement of frequency, but is the subcarrier offset within the analyzed packet burst.

`<x>`, which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If `<x>` is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:FREQUency <NRf>
```

```
CALCulate:WLAN:PERRor:MARKer<x>:FREQUency?
```

## Arguments

Floating point number that indicates the desired subcarrier or frequency to which the marker should be moved. Subcarriers are traditionally numbered with both positive and negative offsets from the CF for the modulation scheme. Valid integral values range from `-maxsubcarrier` to `maxsubcarrier`, where the actual values depend on the modulation standard. Specifying a value of `<NRf>` outside of the range of subcarriers present in the burst selects the nearest valid subcarrier. Subcarriers can also be specified by frequency (also determined by modulation standard).

## Returns

Returns a unitless number representing the subcarrier number or frequency value in Hz.

## Examples

`CALC:WLAN:PERR:MARK2:FREQ 2.0000` sets the position of Marker 2 (M2) to Subcarrier 2 (assuming the units have been set to Subcarrier).

`CALC:WLAN:PERR:MARK2:FREQ?` might return `2.41575E+9`, indicating that Marker 2 (M2) is positioned at 2.41575 MHz.

## CALCulate:WLAN:PERRor:MARKer<x>:TIME

Sets the time value (offset) at the selected marker position on the points trace on the WLAN Phase Error display. The units are either Symbols or Seconds. To set/query the units, use the command [SENSe]:WLAN:UNIT:TIME.

Note that TIME is a measurement of neither absolute nor elapsed time, but symbol offset within the analyzed packet burst.

<x>, which is optional, is the desired marker to use. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Commanding a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:TIME <Nrf>
```

```
CALCulate:WLAN:PERRor:MARKer<x>:TIME?
```

### Arguments

Floating point number that indicates the 0 based symbol or time offset from the start of packet burst. Valid integral values range from 0 to the number of symbols in the burst. Specifying a value outside of the range of symbols present in the burst selects the nearest valid symbol.

### Returns

Returns a unitless integer value in floating point format that represents the symbol associated with the position of the indicated marker, or the time offset.

### Examples

`CALC:WLAN:PERR:MARK3:TIME 362.42E-6` moves Marker 3 (M3) to the 362.42µS position on the trace (assuming the units have been set to Seconds).

`CALC:WLAN:PERR:MARK3:TIME?` might return 5.0000 indicating that Marker 3 is located at symbol 5 (assuming units have been set to Symbols).

## CALCulate:WLAN:PERRor:MARKer<x>:VALue? (Query Only)

Queries the Phase Error value at the selected marker position on the points trace of the WLAN Phase Error display.

The Phase Error value of a marker on the average trace is queried by using the and `CALCulate:WLAN:PERRor:MARKer<x>:AVERage:SYMBol?` queries.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PERRor:MARKer<x>:VALue?
```

## Returns

Returns the Phase Error value at the selected marker position on the WLAN Phase Error display.

## Examples

`CALC:WLAN:PERR:MARK3:VAL?` might return 11.9, indicating a Phase Error value of 11.9 percent for Marker 3 (M3).

## CALCulate:WLAN:PVTime:MARKer<x>:DELTA:X? (Query Only)

Returns the delta time value at the selected marker position in the WLAN Power vs. Time display.

<x> is the desired marker to query. Marker 0 is not valid for the DELTA queries. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:DELTA:X?
```

## Returns

Floating point number that represents the delta time value at the selected marker position.

## Examples

`CALC:WLAN:PVT:MARK2:DELTA:X?` might return 1.28E-6 indicating that the delta time value for Marker 2 (M2) is 1.28  $\mu$ s.

## CALCulate:WLAN:PVTime:MARKer<x>:DELTA:Y? (Query Only)

Queries the vertical delta value for the points trace at the selected marker position on the WLAN Power vs. Time display, using the currently selected Amplitude unit. Set/query the amplitude unit with the command `SENSe:POWer:UNITs`.



<x> is the desired marker to query. Marker 0 is not valid for the DELTa queries. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. Valid integral values range from 1 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:DELTA:Y?
```

### Returns

Returns the vertical delta value for the points trace in the current Amplitude units at the selected marker position

### Examples

`CALC:WLAN:PVT:MARK3:DELTA:Y?` might return `-12.91`, indicating a vertical delta value of `-12.91` dB at the Marker 3 (M3) position.

## **CALCulate:WLAN:PVTime:MARKer<x>:MAXimum (No Query Form)**

Positions the specified marker at the maximum point on the trace in the WLAN Power vs. Time display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:MAXimum
```

### Arguments

None

### Examples

`CALC:WLAN:PVT:MARK2:MAX` positions the Marker 2 (M2) at the maximum point on the trace.

## **CALCulate:WLAN:PVTime:MARKer<x>:PEAK:HIGHer (No Query Form)**

Moves the specified marker to the next peak higher in amplitude on the trace on the WLAN Power vs. Time display.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:HIGHer
```

### **Arguments**

None

### **Examples**

`CALC:WLAN:PVT:MARK2:PEAK:HIGHer` moves Marker 2 (M2) to the next peak higher in amplitude on the trace.

## **CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LEFT (No Query Form)**

Moves the specified marker to the next peak on the left on the trace in the WLAN Power vs. Time display.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Calculate commands

### **Syntax**

```
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LEFT
```

### **Arguments**

None

### **Examples**

`CALC:WLAN:PVT:MARK2:PEAK:LEFT` moves Marker 2 (M2) to the next peak on the left on the trace in the WLAN Power vs. Time display.

## **CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LOWer (No Query Form)**

Moves the marker to the next peak lower in amplitude on the trace on the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:LOWer
```

## Arguments

None

## Examples

`CALC:WLAN:PVT:MARK2:PEAK:LOW` moves Marker 2 (M2) to the next peak lower on the trace.

## **CALCulate:WLAN:PVTime:MARKer<x>:PEAK:RIGHT (No Query Form)**

Moves the marker to the next peak to the right on the trace on the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:PEAK:RIGHT
```

## Arguments

None

## Examples

`CALC:WLAN:PVT:MARK2:PEAK:RIGH` moves Marker 2 (M2) to the next peak on the right on the trace.

## **CALCulate:WLAN:PVTime:MARKer<x>:X**

Sets or queries the value of the horizontal position (in seconds) for the selected marker on the points trace in the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:X <NRF>
```

```
CALCulate:WLAN:PVTime:MARKer<x>:X?
```

## Arguments

Floating point number that specifies the value of the horizontal position of the marker. Range: Start to Stop time (left to right edge of the horizontal axis). Using an out-of-range value causes an execution error (-222, "Data out of range").

## Examples

```
CALC:WLAN:PVT:MARK3:X 1E-6 places Marker 3 at 1  $\mu$ s on the trace.
```

## CALCulate:WLAN:PVTime:MARKer<x>:Y? (Query Only)

Returns the amplitude (vertical position) at the selected marker position in the WLAN Power vs. Time display, using the currently selected Amplitude units. You can set/query the Amplitude units with the command SENSE:POWER:UNITS.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:PVTime:MARKer<x>:Y?
```

## Returns

Floating point number that indicates the amplitude at the selected marker position in the currently selected Amplitude units.

## Examples

```
CALC:WLAN:PVT:MARK1:Y? might return 33.393856E-3 indicating the amplitude at the Marker 1 position is 0.033 dBm.
```

## CALCulate:WLAN:TABLE:MARKer<x>:FREQuency

Assigns or queries the frequency value of the marker in the WLAN Symbol Table. The units are either Frequency in Hz or Subcarrier number. To set the units, use the command [SENSE]:WLAN:UNIT:FREQuency.

Note that FREQuency is not an actual measurement of frequency, but is the subcarrier offset within the analyzed packet burst.

<x>, which is optional, is the desired marker to assign. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:TABLE:MARKer<x>:FREQuency <NRf>
CALCulate:WLAN:TABLE:MARKer<x>:FREQuency?
```

## Arguments

Floating point number that indicates the desired subcarrier or frequency to which the marker should be assigned. Subcarriers are traditionally numbered with both positive and negative offsets from the CF for the modulation scheme. Valid integral values range from  $-\text{maxsubcarrier}$  to  $\text{maxsubcarrier}$ , where the actual values depend on the modulation standard. Specifying a value of <NRf> outside of the range of subcarriers present in the burst selects the nearest valid subcarrier. Subcarriers can also be specified by frequency (also determined by modulation standard).

## Returns

Returns a unitless number representing the subcarrier number or frequency in Hz.

## Examples

`CALC:WLAN:TABLE:MARK3:FREQ 2.0000` assigns Subcarrier 2 to Marker 3 (M3) (assuming the units have been set to Subcarrier).

`CALC:WLAN:TABLEM:MARK2:FREQ?` might return `2.00000E+9`, indicating that Marker 2 (M2) is assigned 2.00 MHz.

## CALCulate:WLAN:TABLE:MARKer<x>:TIME

Sets or queries the vertical value at the selected marker position on the WLAN Symbol Table. The units are either Symbols or Seconds. To set/query the units, use the command [SENSe]:WLAN:UNIT:TIME.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Calculate commands

## Syntax

```
CALCulate:WLAN:TABLE:MARKer<x>:TIME <NRf>
```

`CALCulate:WLAN:TABLE:MARKer<x>:TIME?`

### Arguments

Floating point number that indicates the 0 based symbol or time offset from the start of packet burst. Valid integral values range from 0 to the number of symbols in the burst. Specifying a value outside of the range of symbols present in the burst selects the nearest valid symbol.

### Returns

Returns a unitless integer value in floating point format that represents the symbol associated with the indicated marker, or the time offset.

### Examples

`CALC:WLAN:TABLE:MARK3:TIME 362.42E-6` assigns 362.42  $\mu$ s to Marker 3 (M3) (assuming the units have been set to Seconds).

`CALC:WLAN:TABLE:MARK3:TIME?` might return 5.0000 indicating that symbol 5 is assigned to Marker 3.

## **CALCulate:WLAN:TABLE:MARKer<x>:VALue? (Query Only)**

Queries the decoded symbol value at the selected marker position on the WLAN Symbol Table display.

<x>, which is optional, is the desired marker to query. 0 indicates MR, the reference marker. 1, 2, 3, etc. indicate markers M1, M2, M3, etc. If <x> is not supplied, the command works on the currently selected marker. Valid integral values range from 0 to the number of markers active for the display. Querying a marker which does not exist results in an execution error.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Calculate commands

### Syntax

`CALCulate:WLAN:TABLE:MARKer<x>:VALue?`

### Returns

Returns the decoded symbol value at the selected marker position on the WLAN Symbol Table display.

### Examples

`CALC:WLAN:MERR:MARK3:VAL?` might return 11.9, indicating a decoded symbol value of 11.9 for Marker 3 (M3).

## **Calibration commands**

### **CALibration:ABORt (No query form)**

Aborts any actions related to the alignments in progress.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

```
CALibration:ABORT
```

## Arguments

None

## Examples

`CALibration:ABORT` aborts any actions related to the alignments in progress.

## CALibration:AUTO

Sets or queries whether or not to run alignments automatically.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

```
CALibration:AUTO { OFF | ON | 0 | 1 }
```

```
CALibration:AUTO?
```

## Arguments

OFF or 0 runs alignments on user request.

Use the \*CAL command to perform alignments.

ON or 1 runs alignments as needed without user intervention.

You have to restart measurement if interrupted.

## Examples

`CALibration:AUTO ON` runs alignments automatically as needed.

## CALibration:CORRection:EXTeRnal:EDIT<x>:LABEl

Sets or queries the name of the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.



**Note:** This command has been replaced with `INPut:CORRection:EXTernal:EDIT<x>:LABel`.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

`CALibration:CORRection:EXTernal:EDIT<x>:LABel <name>`

`CALibration:CORRection:EXTernal:EDIT<x>:LABel?`

## Arguments

`<name>::=<string>` specifies the name of the external loss table.

## Examples

`CALibration:CORRection:EXTernal:EDIT<x>:LABel? 'Sample Table 1'` names the External Loss Table 1 "Sample Table 1".

## **CALibration:CORRection:EXTernal:EDIT<x>:NEW (No query form)**

Creates a new external loss table.

The parameter `<x>` = 1 to 3 represent the External Loss Table 1 to 3, respectively.



**Note:** This command has been replaced with `INPut:CORRection:EXTernal:EDIT<x>:NEW`.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

`CALibration:CORRection:EXTernal:EDIT<x>:NEW`

`<freq(1)>, <loss(1)>, <freq(2)>, <loss(2)>, ..., <freq(n)>, <loss(n)>`

## Arguments

`<freq(n)>, <loss(n)>` specifies a pair of frequency (`<NR3>` in Hz) and loss (`<NR3>` in dB) in the external loss table.

The setting range is:

- RSA5103B – 0 to 3 GHz
- RSA5115B – 0 to 15 GHz



- RSA5126B – 0 to 26.5 GHz

Loss: -50 to +80 dB. (A negative value means a gain.)

## Examples

`CALibration:CORRection:EXTErnal:EDIT<x>:NEW 1.0E+9,2.2, 1.5E+9,2.3` creates the External Loss Table 1 specifying the loss of 2.2 dB and 2.3 dB at the frequency of 1 GHz and 1.5 GHz, respectively.

## CALibration:CORRection:EXTErnal:EDIT<x>:STATe

Determines whether to enable or disable the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.



**Note:** This command has been replaced with `INPut:CORRection:EXTErnal:EDIT<x>:STATe`.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

`CALibration:CORRection:EXTErnal:EDIT<x>:STATe { OFF | ON | 0 | 1 }`

`CALibration:CORRection:EXTErnal:EDIT<x>:STATe?`

## Arguments

OFF or 0 disables the external loss table.

ON or 1 enables the external loss table.



**Note:** You can enable one or more tables at the same time.

## Examples

`CALibration:CORRection:EXTErnal:EDIT<x>:STATe? ON` enables the External Loss Table 3.

## CALibration:CORRection:EXTErnal:GAIN[:MAGNitude]

Sets or queries the external gain value. It can be enabled or disabled using the `INPut:CORRection:EXTErnal:GAIN:STATe` command.



**Note:** This command has been replaced with `INPut:CORRection:EXTErnal:GAIN[:MAGNitude]`.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

```
CALibration:CORRection:EXTErnal:GAIN[:MAGNitude] <value>
```

```
CALibration:CORRection:EXTErnal:GAIN[:MAGNitude]?
```

## Arguments

<value> ::= <NRf> specifies the external gain value.

Range: -50 to +30 dB.

## Examples

```
CALibration:CORRection:EXTErnal:GAIN[:MAGNitude]? -10 sets the external gain to -10 dB.
```

## CALibration:CORRection:EXTErnal:GAIN:STATE

Determines whether to enable or disable the external gain value.



**Note:** This command has been replaced with `INPut:CORRection:EXTErnal:GAIN[:MAGNitude]`.

## Conditions

Measurement views: All

## Group

Calibration commands

## Syntax

```
CALibration:CORRection:EXTErnal:GAIN:STATE { OFF | ON | 0 | 1 }
```

```
CALibration:CORRection:EXTErnal:GAIN:STATE?
```

## Related Commands

```
INPut:CORRection:EXTErnal:GAIN[:MAGNitude]
```

## Arguments

OFF or 0 disables the external gain value.

ON or 1 enables the external gain value.

## Examples

```
CALibration:CORRection:EXTErnal:GAIN:STATE ON enables the external gain value.
```

## CALibration:CORRection:EXTernal:PROBe:CONNect? (Query only)

Queries whether the external probe is connected to the analyzer or not.

### Conditions

Measurement views: All

### Group

Calibration commands

### Syntax

```
CALibration:CORRection:EXTernal:PROBe:CONNect?
```

### Arguments

None

### Returns

```
{ 0 | 1 }
```

0 indicates that the external probe is not connected to the analyzer.

1 indicates that the external probe is connected to the analyzer.

### Examples

CALibration:CORRection:EXTernal:PROBe:CONNect? might return ON, indicating that the external probe is connected to the analyzer.

## CALibration:CORRection:EXTernal:PROBe[:MAGNitude]? (Query only)

Queries the external probe attenuation.

### Conditions

Measurement views: All

### Group

Calibration commands

### Syntax

```
CALibration:CORRection:EXTernal:PROBe[:MAGNitude]?
```

### Arguments

None

### Returns

<attenuation>::=<NRf> The probe attenuation value in dB.

## Examples

`CALibration:CORRection:EXTErnal:PROBe[:MAGNitude]?`  might return 10, indicating that the probe attenuation is 10 dB.

## **CALibration:CORRection:EXTErnal:PROBe:STATe**

Determines whether or not to correct data for the external probe attenuation.

### Conditions

Measurement views: All

### Group

Calibration commands

### Syntax

```
CALibration:CORRection:EXTErnal:PROBe:STATe { OFF | ON | 0 | 1 }
```

```
CALibration:CORRection:EXTErnal:PROBe:STATe?
```

### Arguments

OFF or 0 does not correct data for the external probe attenuation.

ON or 1 corrects data for the external probe attenuation.

### Examples

`CALibration:CORRection:EXTErnal:PROBe:STATe ON`  corrects data for the external probe attenuation.

## **CALibration:CORRection:EXTErnal:TYPE**

Sets or queries the data type to apply the external loss table corrections.



**Note:** This command has replaced with `INPut:CORRection:EXTErnal:TYPE`

### Conditions

Measurement views: All

### Group

Calibration commands

### Syntax

```
CALibration:CORRection:EXTErnal:TYPE { TRACe | DATA }
```

```
CALibration:CORRection:EXTErnal:TYPE
```

### Related Commands

[INPut:CORRection:EXTErnal:GAIN:STATe](#)

## Arguments

TRACe selects traces in the Spectrum, Spectrogram, Spurious, and Amplitude versus Time views. Selecting TRACe disables the `CALibration:CORRection:EXTErnal:GAIN:STATe` command.

DATA selects all acquired data. Selecting DATA enables the `CALibration:CORRection:EXTErnal:GAIN:STATe` command.

## Examples

`CALibration:CORRection:EXTErnal:TYPE TRACe` selects traces to apply the external loss table corrections.

## CALibration:TXGain:ABORt (No query form)

Cancels a calibration.

## Conditions

Measurement view: Transmission Gain, Perform Calibration window

## Group

Calibration commands

## Syntax

```
CALibration:TXGain:ABORt
```

## Arguments

None

## Examples

`CALibration:TXGain:ABORt` cancels a calibration that is in process.

## CALibration:TXGain[:ALL] (No query form)

Performs a calibration using the current settings.

## Conditions

Measurement view: Transmission Gain, Perform Calibration window

## Group

Calibration commands

## Syntax

```
CALibration:TXGain[:ALL]
```

## Arguments

None

## Examples

`CALibration:TXGain[:ALL]` performs a calibration using the current settings.

## **CALibration:TXGain:FINish? (Query only)**

Queries whether or not the calibration is completed.

## Conditions

Measurement view: Transmission Gain, Perform Calibration window

## Group

Calibration commands

## Syntax

`CALibration:TXGain:FINish?`

## Returns

0 indicates calibration is in progress.

1 indicates calibration is completed.

## Examples

`CALibration:TXGain:FINish?` might return 1, indicating that the calibration is completed.

## **CALibration:TXGain:STATe? (Query only)**

Queries the calibration state for the Transmission Gain measurement.

## Conditions

Measurement view: Transmission Gain

## Group

Calibration commands

## Syntax

`CALibration:TXGain:STATe?`

## Returns

CALIBRATED means that the unit has been calibrated and is making measurements at the same frequencies at which it was calibrated.

CALIBRATED? means that the unit has been calibrated, but one or more of the operating conditions has changed. Operating conditions are Tracking Generator Output Power, Reference Level, Preamplifier On/Off, RF Attenuation, or Temperature (>5 °C from the calibrated temperature).

INTERPOLATED means that the unit has been calibrated, but is making measurements at frequencies other than the calibration points and is interpolating the calibration between points.

## Examples

`CALibration:TXGain:STATe?` might return `CALIBRATED`, indicating that the measurement has been calibrated with the current settings.

## \*CLS (No Query Form)

Clears the analyzer status data structures. Refer to Section 3, *Status and Events*, for the register information.

The \*CLS command clears the following

- the Event Queue
- the Standard Event Status Register (SESR)
- the Status Byte Register (except the MAV bit; see below)

If the \*CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates information is in the output queue. The device clear (DCL) GPIB control message will clear the output queue and thus MAV. \*CLS does not clear the output queue or MAV. (A complete discussion of these registers and bits, and of event handling in general is described in the *Status and Events* section)

\*CLS can suppress a Service Request that is to be generated by an \*OPC. This will happen if a hardcopy output or single sequence acquisition operation is still being processed when the \*CLS command is executed.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

\*CLS

## Related Commands

\*ESE, \*ESR?, \*SRE, \*STB?

## Arguments

None

## Examples

\*CLS clears the analyzer status data structures.

## Display commands

### DISPlay:ACPower:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Channel power and ACPR view.

## Conditions

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

```
DISPlay:ACPower:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }  
DISPlay:ACPower:MARKer:SHOW:STATe?
```

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

DISPLAY:ACPOWER:MARKER:SHOW:STATEON shows the readout for the selected marker in the graph.

## DISPlay:ACPower:PLEVel:SHOW:STATe

Determines whether to show or hide the power levels in the Channel power and ACPR view.

## Conditions

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

```
DISPlay:ACPower:PLEVel:SHOW:STATe { OFF | ON | 0 | 1 }  
DISPlay:ACPower:PLEVel:SHOW:STATe?
```

## Arguments

OFF or 0 hides the power levels in the graph.

ON or 1 shows the power levels in the graph.

## Examples

DISPLAY:ACPOWER:PLEVEL:SHOW:STATEON shows the power levels in the graph.

## DISPlay:ACPower:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Channel power and ACPR view.

Vertical offset = Reference level, Vertical scale = 100 dB,

Horizontal offset = Center frequency, and Horizontal scale = Default span



**Conditions**

Measurement views: Channel power and ACPR

**Group**

Display commands

**Syntax**

```
DISPlay:ACPower:RESet:SCALE
```

**Arguments**

None

**Examples**

DISPLAY:ACPOWER:RESET:SCALE resets the horizontal and vertical scale to the default values.

**DISPlay:ACPower:WINDow:TRACe:GRATICule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen.

**Conditions**

Measurement views: Channel power and ACPR

**Group**

Display commands

**Syntax**

```
DISPlay:ACPower:WINDow:TRACe:GRATICule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:ACPower:WINDow:TRACe:GRATICule:GRID:STATe?
```

**Arguments**

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

**Examples**

DISPLAY:ACPOWER:WINDOW:TRACE:GRATICULE:GRID:STATEON shows the graticule grid on the screen.

**DISPlay:ACPower:X[:SCALE]**

Sets or queries the horizontal range of the Channel power and ACPR graph.

**Conditions**

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

`DISPlay:ACPower:X[:SCALE] <value> DISPlay:ACPower:X[:SCALE]?`

## Related Commands

[\*DISPlay:ACPower:X\[:SCALE\]:OFFSet\*](#)

## Arguments

`<value> ::= <NRf>` specifies the horizontal range.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`DISPLAY:ACPOWER:X:SCALE 10MHz` sets the horizontal range to 10 MHz.

## DISPlay:ACPower:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Channel power and ACPR view.

## Conditions

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

`DISPlay:ACPower:X[:SCALE]:AUTO`

## Arguments

None

## Examples

`DISPLAY:ACPOWER:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:ACPower:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Channel power and ACPR graph.

## Conditions

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

```
DISPlay:ACPower:X[:SCALE]:OFFSet <value> DISPlay:ACPower:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:ACPower:X\[:SCALE\]](#)

## Arguments

<value>::=<NRf> specifies the minimum horizontal value.

Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

## Examples

DISPlay:ACPower:X:SCALE:OFFSet 1.45GHz sets the minimum horizontal value to 1.45 GHz in the Channel power and ACPR graph.

## DISPlay:ACPower:Y[:SCALE]

Sets or queries the vertical range of the channel power and ACPR graph.

## Conditions

Measurement views: Channel power and ACPR

## Group

Display commands

## Syntax

```
DISPlay:ACPower:Y[:SCALE] <value> DISPlay:ACPower:Y[:SCALE]?
```

## Related Commands

[DISPlay:ACPower:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::=<NRf> specifies the vertical range. Range: 0.1 to 200 dB.

## Examples

DISPLAY:ACPOWER:Y:SCALE 100 sets the vertical range to 100 dB in the Channel power and ACPR graph.

## **DISPlay:ACPower:Y[:SCALE]:AUTO (No Query Form)**

Rescales the vertical axis automatically to fit the waveform to the screen in the Channel power and ACPR view.

### **Conditions**

Measurement views: Channel power and ACPR

### **Group**

Display commands

### **Syntax**

```
DISPlay:ACPower:Y[:SCALE]:AUTO
```

### **Arguments**

None

### **Examples**

DISPLAY:ACPOWER:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## **DISPlay:ACPower:Y[:SCALE]:OFFSet**

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Channel power and ACPR graph.

### **Conditions**

Measurement views: Channel power and ACPR

### **Group**

Display commands

### **Syntax**

```
DISPlay:ACPower:Y[:SCALE]:OFFSet <value> DISPlay:ACPower:Y[:SCALE]:OFFSet?
```

### **Related Commands**

[DISPlay:ACPower:Y\[:SCALE\]](#)

### **Arguments**

<value> ::= <NRf> specifies the vertical offset. Range: -170 to +50 dBm.

### **Examples**

DISPLAY:ACPOWER:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBm in the Channel power and ACPR graph.

## DISPlay:ADEMod:MEASview:DELeTe (No Query Form)

Deletes the measurement view in the general purpose analog demodulation measurements.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

```
DISPlay:ADEMod:MEASview:DELeTe { AM | FM | PM }
```

### Arguments

The following table lists the arguments.

**Table 33: Analog demodulation measurement views**

Argument	View
AM	AM measurement (modulation factor versus time)
FM	FM measurement (frequency deviation versus time)
PM	PM measurement (phase deviation versus time)

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement is not running") will be returned.

### Examples

DISPLAY:ADEMOD:MEASVIEW:DELETEAM deletes the AM measurement view.

## DISPlay:ADEMod:MEASview:NEW (No Query Form)

Displays a new measurement view in the general purpose analog demodulation measurements.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

```
DISPlay:ADEMod:MEASview:NEW { AM | FM | PM }
```

### Arguments

(See Table 32 on page 2-481.) If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

`DISPLAY:ADEMOD:MEASVIEW:NEWAM` creates the AM measurement view.

## DISPlay:ADEMod:MEASview:SElect

Selects a measurement view on the screen in the general purpose analog demodulation measurements. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the `DISPlay:WINDow:OPTimized:MEASurement?` query.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Display commands

## Syntax

```
DISPlay:ADEMod:MEASview:SElect { AM | FM | PM } DISPlay:ADEMod:MEASview:SElect?
```

## Arguments

(See Table 32 on page 2-481.) If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

`DISPLAY:ADEMOD:MEASVIEW:SELECTAM` selects the AM measurement view.

## DISPlay:{AM|FM|PM}:MARKer:SHOW:STATe

Determines whether to show or hide the marker readout in the AM/FM/PM measurement view.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Display commands

## Syntax

```
DISPlay:{AM|FM|PM}:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }  
DISPlay:{AM|FM|PM}:MARKer:SHOW:STATe?
```

## Arguments

`OFF` or `0` hides the marker readout on the screen.

`ON` or `1` shows the marker readout on the screen.

## Examples

`DISPLAY:AM:MARKER:SHOW:STATE ON` shows the marker readout in the AM measurement view.

## **DISPlay:{AM|FM|PM}:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

`DISPlay:{AM|FM|PM}:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }`

`DISPlay:{AM|FM|PM}:WINDow:TRACe:GRATicule:GRID:STATe?`

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

`DISPLAY:AM:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the AM measurement view.

## **DISPlay:{AM|FM|PM}:X:RSCale (No Query Form)**

Rescales the horizontal axis automatically to fit the waveform to the screen in the AM/FM/PM measurement display.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

`DISPlay:{AM|FM|PM}:X:RSCale`

### Arguments

None

## Examples

`DISPLAY:AM:X:RSCALE` rescales the horizontal scale automatically to fit the

waveform to the screen.

## **DISPlay:{AM|FM|PM}:X[:SCALE]:AUTO (No Query Form)**

Sets the horizontal scale automatically to fit the waveform to the screen in the AM/FM/PM measurement display.

### **Conditions**

Measurement views: General purpose analog demodulation

### **Group**

Display commands

### **Syntax**

```
DISPlay:{AM|FM|PM}:X[:SCALE]:AUTO
```

### **Arguments**

None

### **Examples**

DISPLAY:AM:X:SCALE:AUTO sets the horizontal scale automatically to fit the waveform to the screen in the AM/FM/PM measurement display.

## **DISPlay:{AM|FM|PM}:X[:SCALE]:FULL**

Sets or queries the horizontal scale (full-scale time) of the AM/FM/PM measurement graph.

### **Conditions**

Measurement views: General purpose analog demodulation

### **Group**

Display commands

### **Syntax**

```
DISPlay:{AM|FM|PM}:X[:SCALE]:FULL <value>
```

```
DISPlay:{AM|FM|PM}:X[:SCALE]:FULL?
```

### **Arguments**

<value> ::= <NRf> specifies the horizontal scale in full-scale time.

Range: 0 to 1027 s.

### **Examples**

DISPlay:AM:X:SCALE:FULL 35us sets the horizontal scale to 35  $\mu$ s in the AM measurement.



## DISPlay:{AM|FM|PM}:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the AM/FM/PM measurement graph.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

```
DISPlay:{AM|FM|PM}:X[:SCALe]:OFFSet <value>
```

```
DISPlay:{AM|FM|PM}:X[:SCALe]:OFFSet?
```

### Arguments

<value>::=<NRf> specifies the minimum horizontal value.

Range:  $-0.9 \times (\text{horizontal scale})$  to  $+0.9 \times (\text{horizontal scale})$

### Examples

DISPLAY:AM:X:SCALE:OFFSET 20.075us sets the minimum horizontal value to 20.075  $\mu$ s.

## DISPlay:{AM|FM|PM}:Y:RSCale (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the AM/FM/PM measurement display.

### Conditions

Measurement views: General purpose analog demodulation

### Group

Display commands

### Syntax

```
DISPlay:{AM|FM|PM}:Y:RSCale
```

### Arguments

None

### Examples

DISPLAY:AM:Y:RSCALE rescales the vertical scale automatically to fit the waveform to the screen in the AM measurement.

## DISPlay:{AM|FM|PM}:Y[:SCALe]

Sets or queries the vertical range of the AM/FM/PM measurement graph.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Display commands

## Syntax

`DISPlay:{AM|FM|PM}:Y[:SCALE] <value> DISPlay:{AM|FM|PM}:Y[:SCALE]?`

## Arguments

`<value>::=<NRf>` specifies the vertical range.

The setting range depends on measurements as shown in the following table.

**Table 34: Setting range**

Measurement	Setting range
AM	1% to 500%
FM	10 Hz to 120 MHz
PM	10° to 400°

## Examples

`DISPLAY:AM:Y:SCALE 75` sets the vertical range to 75% in the AM measurement graph.

## DISPlay:{AM|FM|PM}:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) in the AM/FM/PM measurement graph.

## Conditions

Measurement views: General purpose analog demodulation

## Group

Display commands

## Syntax

`DISPlay:{AM|FM|PM}:Y[:SCALE]:OFFSet <value> DISPlay:{AM|FM|PM}:Y[:SCALE]:OFFSet?`

## Arguments

`<value>::=<NRf>` specifies the vertical offset.

The setting range depends on measurements as shown in the following table.

**Table 35: Setting range**

Measurement	Setting range
AM	-500% to +500%
FM	$-(\text{maxVertScale} * (1 + 1.1) * 0.5)$ to $+(\text{maxVertScale} * (1 + 1.1) * 0.5)$
PM	-180° to +180°

## Examples

`DISPLAY:AM:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5% in the AM measurement graph.

## DISPlay:ANTenna:AZIMuth:SHOW:STATe

The command checks the Show Compass box in the UI to display the compass. The query form returns whether or not the Show Compass box is checked.

## Group

Display commands

## Syntax

`DISPlay:ANTenna:AZIMuth:SHOW:STATe { 1 | 0 | ON | OFF }`

`DISPlay:ANTenna:AZIMuth:SHOW:STATe?`

## Arguments

1 or ON sets compass to show.

0 or OFF sets compass to hide.

## Returns

1 or ON means compass is set to show.

0 or OFF means compass is set to hide.

## DISPlay:AUDio:MEASview:DELeTe (No Query Form)

Deletes the specified audio analysis view.

## Conditions

Measurement views: Audio Spectrum, Audio Summary

## Group

Display commands

## Syntax

```
DISPlay:AUDio:MEASview:DELeTe { SPECTrum | SUMMary }
```

## Arguments

SPECTrum deletes the Audio Spectrum view.

SUMMary deletes the Audio Summary view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

```
DISPLAY:AUDIO:MEASVIEW:DELETE SUMMary deletes the Audio Summary view.
```

## DISPlay:AUDio:MEASview:NEW (No Query Form)

Displays a new audio analysis view.

## Conditions

Measurement views: Audio Spectrum, Audio Summary

## Group

Display commands

## Syntax

```
DISPlay:AUDio:MEASview:NEW { SPECTrum | SUMMary }
```

## Arguments

SPECTrum creates a new Audio Spectrum view.

SUMMary creates a new Audio Summary view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

```
DISPLAY:AUDIO:MEASVIEW:NEW SUMMary creates a new Audio Summary view.
```

## DISPlay:AUDio:MEASview:SELEct

Selects an audio analysis view on the screen. The query returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

## Conditions

Measurement views: Audio Spectrum, Audio Summary

## Group

Display commands

## Syntax

```
DISPlay:AUDio:MEASview:SElect { SPECTrum | SUMMary }
```

```
DISPlay:AUDio:MEASview:SElect?
```

## Arguments

`SPECTrum` creates a new Audio Spectrum view.

`SUMMary` creates a new Audio Summary view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

```
DISPLAY:AUDIO:MEASVIEW:SELECT SPECTrum selects the Audio Spectrum  
display.
```

## DISPlay:AUDio:SPECTrum:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:FREQuency:AUTO
```

## Arguments

None

## Examples

```
DISPLAY:AUDIO:SPECTRUM:FREQUENCY:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.
```

## DISPlay:AUDio:SPECTrum:FREQuency:[SCALE]:START

Sets or queries the start frequency (left edge) of the audio spectrum graph.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:FREQuency:[SCALe]:START <value>
```

```
DISPlay:AUDio:SPECTrum:FREQuency:[SCALe]:START?
```

## Related Commands

[\*DISPlay:AUDio:SPECTrum:FREQuency:\[SCALe\]:STOP\*](#)

## Arguments

<value> ::= <NRf> specifies the start frequency.

## Examples

DISPLAY:AUDIO:SPECTRUM:FREQUENCY:[SCALE]:START 0 sets the start frequency to 0 Hz.

## DISPlay:AUDio:SPECTrum:FREQuency:[SCALe]:STOP

Sets or queries the stop frequency (right edge) of the audio spectrum graph.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:FREQuency:[SCALe]:STOP <value>
```

```
DISPlay:AUDio:SPECTrum:FREQuency:[SCALe]:STOP?
```

## Related Commands

[\*DISPlay:AUDio:SPECTrum:FREQuency:\[SCALe\]:START\*](#)

## Arguments

<value> ::= <NRf> specifies the stop frequency.

## Examples

DISPLAY:AUDIO:SPECTRUM:FREQUENCY:[SCALE]:STOP 20E+3 sets the stop frequency to 20 kHz.

## DISPlay:AUDio:SPECTrum:MARKer:SHOW:STATe

Determines whether to show or hide the readouts for the selected marker in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:AUDio:SPECTrum:MARKer:SHOW:STATe
```

## Arguments

OFF or 0 hides the readout for the selected marker in the view.

ON or 1 shows the readout for the selected marker in the view.

## Examples

DISPLAY:AUDIO:SPECTRUM:MARKER:SHOW:STATE ON shows the readout for the selected marker in the audio spectrum view.

## DISPlay:AUDio:SPECTrum:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scales in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:RESet:SCALE
```

## Arguments

None

## Examples

DISPLAY:AUDIO:SPECTRUM:RESET:SCALE resets the horizontal and vertical scales.

## DISPlay:AUDio:SPECTrum:SCALE:LOG:STATe

Determines whether or not to set the horizontal axis logarithmic in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:SCALE:LOG:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:AUDio:SPECTrum:SCALE:LOG:STATe?
```

## Arguments

OFF or 0 sets the horizontal axis linear.

ON or 1 sets the horizontal axis logarithmic.

## Examples

DISPLAY:AUDIO:SPECTRUM:SCALE:LOG:STATE ON sets the horizontal axis logarithmic in the audio spectrum view.

## DISPlay:AUDio:SPECTrum:SHOW:NHARmonic:THReshold

Sets or queries the appearance of the non-harmonic threshold in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:SHOW:NHARmonic:THReshold { SHADed | LINE | OFF }
```

```
DISPlay:AUDio:SPECTrum:SHOW:NHARmonic:THReshold?
```

## Arguments

SHADed uses shading for the non-harmonic threshold area in the display.

LINE uses a line for the non-harmonic threshold area.

OFF turns off the non-harmonic threshold in the display.

## Examples

DISPLAY:AUDIO:SPECTRUM:SHOW:NHARMONIC:THRESHOLD SHADed uses shading to identify the non-harmonic threshold area in the audio spectrum view.

## DISPlay:AUDio:SPECTrum:TABLE:SHOW:STATe

Determines whether to show or hide the results table in the audio spectrum view.



## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:TABLE:SHOW:STATe { OFF | ON | 0 | 1 } DISPlay:AUDio:SPECTrum:TABLE:SHOW:STATe?
```

## Arguments

OFF or 0 disables (hides) the results table in the audio spectrum view.

ON or 1 shows the results table in the audio spectrum view.

## Examples

```
DISPLAY:AUDIO:SPECTRUM:TABLE:SHOW:STATE OFF hides the results table  
in the audio spectrum view.
```

## DISPlay:AUDio:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }  
DISPlay:AUDio:SPECTrum:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

```
DISPLAY:AUDIO:SPECTRUM:WINDOW:TRACE:GRATICULE:GRID:STATE OFF  
hides the graticule grid on the screen.
```

## DISPlay:AUDio:SPECTrum:Y:[SCALE]

Sets or queries the vertical scale in the audio spectrum view. The range of the scale is from 0.1 dB to 200 dB.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:Y:[SCALE] <value> DISPlay:AUDio:SPECTrum:Y:[SCALE]?
```

## Arguments

<value> ::= <Nrf> specifies the vertical scale value in dB.

## Examples

```
DISPLAY:AUDIO:SPECTRUM:Y:[SCALE] 100 sets the vertical scale to 100 dB.
```

## DISPlay:AUDio:SPECTrum:Y:[SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:Y:[SCALE]:AUTO
```

## Arguments

None

## Examples

```
DISPLAY:AUDIO:SPECTRUM:Y:[SCALE]:AUTO rescales the vertical axis automatically.
```

## DISPlay:AUDio:SPECTrum:Y:[SCALE]:OFFSet

Sets or queries the vertical position (offset) in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Display commands

## Syntax

```
DISPlay:AUDio:SPECTrum:Y:[SCALe]:OFFSet <value> DISPlay:AUDio:SPECTrum:Y:
[SCALe]:OFFSet?
```

## Arguments

<value> ::= <NRf> specifies the vertical offset in dBm.

## Examples

DISPLAY:AUDIO:SPECTRUM:Y:[SCALE]:OFFSET 150 sets the vertical offset to 150 dBm2.

## DISPlay:AVTime[:MEASview<y>]:LEGend:STATe

Determines whether to show or hide the trace legend in the amplitude versus time view. The legend indicates the trace detection and function on the screen for each displayed trace.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:LEGend:STATe { OFF | ON | 0 | 1 }
DISPlay:AVTime[:MEASview<y>]:LEGend:STATe?
```

## Arguments

OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

DISPLAY:AVTIME:MEASVIEW1:LEGEND:STATEON shows the trace legend on the screen.

## DISPlay:AVTime[:MEASview<y>]:SELEct:FIRSt

Queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
:DISPlay:AVTime[:MEASview<y>]:SElect:FIRSt
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

{ 0 | 1 }

0 indicates that the results are not being displayed in the readout for the first trace.

1 indicates that the results are being displayed in the readout for the first trace.

## Examples

DISPLAY:AVTIME:MEASVIEW1:SELECT:FIRST? might return 1, indicating that the results are being displayed in the readout for Trace 1 in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:SElect:SECond

Sets or queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
:DISPlay:AVTime[:MEASview<y>]:SElect:SECond
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

{ 0 | 1 }

0 indicates that the results are not being displayed in the readout for the first trace.

1 indicates that the results are being displayed in the readout for the first trace.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:SELECT:FIRST?` might return 1, indicating that the results are being displayed in the readout for Trace 2 in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:SElect:THIRd

Queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

All traces are valid.

### Conditions

Measurement views: Amplitude versus Time

### Group

Trace commands

### Syntax

```
:DISPlay:AVTime[:MEASview<y>]:SElect:THIRd
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

{0|1}

0 indicates that the results are not being displayed in the readout for the first trace.

1 indicates that the results are being displayed in the readout for the first trace.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:SELECT:FIRST?` Might return 1, indicating that the results are being displayed in the readout for Trace 3 in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:SElect:FOURth

Queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

All traces are valid.

### Conditions

Measurement views: Amplitude versus Time

### Group

Trace commands

## Syntax

```
:DISPlay:AVTime[:MEASview<y>]:SElect:FOURth
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
{0|1}
```

0 indicates that the results are not being displayed in the readout for the first trace.

1 indicates that the results are being displayed in the readout for the first trace.

## Examples

DISPLAY:AVTIME:MEASVIEW1:SELECT:FIRST? might return 1, indicating that the results are being displayed in the readout for Trace 4 in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker on the screen in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:AVTime[:MEASview<y>]:MARKer:SHOW:STATe?
```

## Arguments

OFF or 0 hides the readout for the selected marker on screen.

ON or 1 shows the readout for the selected marker on screen.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

DISPLAY:AVTIME:MEASVIEW1:MARKER:SHOW:STATEON shows the readout for the selected marker on screen in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:RESet (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Amplitude versus Time view.

Vertical offset = Reference level, Vertical scale = 100 dB,

Horizontal offset = Analysis offset, and Horizontal scale = Analysis length

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:RESet
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

DISPLAY:AVTIME:MEASVIEW1:RESET resets the horizontal and vertical scale to the default values in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:TRIGger:LEVel:STATe

Determines whether to show or hide the power trigger level line on the screen in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:TRIGger:LEVel:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:AVTime[:MEASview<y>]:TRIGger:LEVel:STATe?
```

## Arguments

OFF or 0 hides the power trigger level line.

ON or 1 shows the power trigger level line.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

See Arguments.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:TRIGGER:LEVEL:STATEON` shows the power trigger level line on the screen in the Amplitude versus Time Measurement View 1.

## **DISPlay:AVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Amplitude versus Time

### Group

Display commands

### Syntax

```
DISPlay:AVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }  
DISPlay:AVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

See Arguments.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:WINDOW:TRACE:GRATICULE:GRID:STATEON` shows the graticule grid on the screen in the Amplitude versus Time Measurement View 1.

## **DISPlay:AVTime[:MEASview<y>]:X:RSCale (No Query Form)**

Rescales the horizontal axis automatically to fit the waveform to the screen in the Amplitude versus Time display.

### Conditions

Measurement views: Amplitude versus Time

### Group

Display commands

### Syntax

```
DISPlay:AVTime[:MEASview<y>]:X:RSCale
```



## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`DISPLAY:AVTIME:MEASVIEW1:X:RSCALE` rescales the horizontal scale automatically to fit the waveform to the screen in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Amplitude versus Time view. Executing this command sets DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe ON.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

`DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO`

## Related Commands

[\*DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe\*](#)

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`DISPlay:AVTime:MEASview<y>:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe?
```

## Arguments

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:FULL](#) and [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#) commands.

ON or 1 specifies that the horizontal scale is set automatically.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

See Arguments.

## Examples

DISPLAY:AVTIME:MEASVIEW1:X:SCALE:AUTO:STATE ON specifies that the horizontal scale is set automatically in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:FULL

Sets or queries the horizontal scale (full-scale time) of the Amplitude versus Time graph. Programming a specified scale sets DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe OFF.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:FULL <value>
```

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:FULL?
```

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe](#), [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

## Arguments

<value>::=<NRf> specifies the horizontal scale in full-scale time. Use the [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:MAXimum?](#) and [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:MINimum?](#) queries to get the upper and lower limits of the setting range.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

`DISPLAY:AVTIME:MEASVIEW1:X:SCALE:FULL 25.6us` sets the horizontal scale to 25.6  $\mu$ s in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

`DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:MAXimum?`

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:FULL](#)

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> The upper limit of the horizontal scale setting range.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:X:SCALE:MAXIMUM?` might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:MINimum?
```

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALe\]:FULL](#)

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> The lower limit of the horizontal scale setting range.

## Examples

DISPLAY:AVTIME:MEASVIEW1:X:SCALE:MINIMUM? might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Amplitude versus Time graph. Programming a specified offset sets DISPlay:AVTime[:MEASview<y>]:X[:SCALe] AUTO:STATe OFF.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet <value>
```

```
DISPlay:AVTime[:MEASview<y>]:X[:SCALe]:OFFSet?
```

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALe\]:AUTO:STATe](#), [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALe\]:FULL](#)

## Arguments

<value>::=<NRf> specifies the minimum horizontal value. Use the

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet:MAXimum?](#) and [DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet:MINimum?](#) queries to get the upper and lower limits of the setting range.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

`DISPLAY:AVTIME:MEASVIEW1:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

### Conditions

Measurement views: Amplitude versus Time

### Group

Display commands

### Syntax

`DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum?`

### Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

<NRf> The upper limit of the horizontal offset setting range.

## Examples

`DISPLAY:AVTIME:MEASVIEW1:X:SCALE:OFFSET:MAXIMUM?` might return -1.812E-3, indicating that the upper limit of the horizontal offset setting range is -1.812 ms in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

### Conditions

Measurement views: Amplitude versus Time

### Group

Display commands

### Syntax

`DISPlay:AVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MINimum?`

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> The lower limit of the horizontal offset setting range.

## Examples

DISPLAY:AVTIME:MEASVIEW1:X:SCALE:OFFSET:MINIMUM? might return -16.28E-3, indicating that the lower limit of the horizontal offset setting range is -16.28 ms in the Amplitude versus Time Measurement View 1.

## DISPlay:AVTime[:MEASview<y>]:Y:RSCale (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Amplitude versus Time display.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:Y:RSCale
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

DISPLAY:AVTIME:MEASVIEW1:Y:RSCALE rescales the vertical scale automatically to fit the waveform to the screen in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:FULL

Sets or queries the vertical range of the Amplitude versus Time graph.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:FULL <value>
```

```
DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:FULL?
```

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::=<NRf> specifies the vertical range. Range: 0.1 to 200 dB.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

DISPLAY:AVTIME:MEASVIEW1:Y:SCALE:FULL 100 sets the vertical range to 100 dB in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Amplitude versus Time graph.

## Conditions

Measurement views: Amplitude versus Time

## Group

Display commands

## Syntax

```
DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:AVTime[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:AVTime\[:MEASview<y>\]:Y\[:SCALE\]:FULL](#)

## Arguments

<value>::=<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf>

## Examples

`DISPLAY:AVTIME:MEASVIEW1:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the Amplitude versus Time Measurement View 1 graph.

## DISPlay:BIBEmissions:MARKer<x>:SHOW:STATE

Sets or queries to show or hide the readout for the selected marker Bluetooth InBand Emission view.

The parameter <x> = 1 to 4; Marker 0 (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command.

### Conditions

Measurement views: Bluetooth InBand Emission

### Group

Display commands

### Syntax

```
DISPlay:BIBEmissions:MARKer<x>:SHOW:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:BIBEmissions:MARKer<x>:SHOW:STATE?
```

### Arguments

`OFF` or `0` hides the readout for the selected marker in the graph.

`ON` or `1` shows the readout for the selected marker in the graph.

### Returns

See Arguments.

## Examples

`DISPLAY:BIBEMISSIONS:MARKER1:SHOW:STATE ON` shows the readout for the Marker1 (M1) in the graph.

## DISPlay:BIBEmissions:PLEVel:SHOW:STATE

Determines whether to show or hide the power levels in the Bluetooth InBand Emission view.

### Conditions

Measurement views: Bluetooth InBand Emission

### Group

Display commands

### Syntax

```
DISPlay:BIBEmissions:PLEVel:SHOW:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:BIBEmissions:PLEVel:SHOW:STATE?
```



## Arguments

OFF or 0 hides the power levels in the graph.

ON or 1 shows the power levels in the graph.

## Returns

See Arguments.

## Examples

`DISPLAY:BIBEMISSIONS:PLEVEL:SHOW:STATEON` shows the power levels in the graph.

## DISPlay:BIBEmissions:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale to the default values in the Bluetooth InBand Emission view.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIBEmissions:RESet:SCALE
```

## Arguments

None

## Examples

`DISPLAY:BIBEMISSIONS:RESET:SCALE` resets the horizontal and vertical scale to the default values.

## DISPlay:BIBEmissions:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule hidden or showing state.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIBEmissions:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | }
```

```
DISPlay:BIBEmissions:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

DISPLAY:BIBEMISSIONS:WINDOW:TRACE:GRATICULE:GRID:STATEON shows the graticule grid on the screen.

## DISPlay:BIBEmissions:X[:SCALE]

Sets or queries the horizontal range of the Bluetooth InBand Emission graph.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

DISPlay:BIBEmissions:X[:SCALE] <value>

DISPlay:BIBEmissions:X[:SCALE]?

## Related Commands

[DISPlay:BIBEmissions:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::=<NRf> specifies the horizontal range.

## Returns

<NRf>

## Examples

DISPLAY:BIBEMISSIONS:X:SCALE 10MHZ sets the horizontal range to 10 MHz.

## DISPlay:BIBEmissions:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Bluetooth InBand Emission view.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIBEmISSIONS:X[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:BIBEMISSIONS:X:SCALE:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:BIBEmISSIONS:X[:SCALE]:OFFSet

Sets or queries the center frequency of the Bluetooth InBand Emission graph.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIBEmISSIONS:X[:SCALE]:OFFSet <value>
```

```
DISPlay:BIBEmISSIONS:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:BIBEmISSIONS:X\[:SCALE\]](#)

## Arguments

<value> ::= <NRf> specifies the minimum horizontal value.

## Returns

<NRf>

## Examples

DISPLAY:BIBEMISSIONS:X:SCALE:OFFSET 1.45GHZ sets the center frequency to 1.45 GHz.

## DISPlay:BIBEmISSIONS:Y[:SCALE]

Sets or queries the vertical range of the Bluetooth InBand Emission graph.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIbEmissions:Y[:SCALE] <value>
```

```
DISPlay:BIbEmissions:Y[:SCALE]?
```

## Related Commands

[DISPlay:BIbEmissions:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value> ::= <NRf> specifies the vertical range. Range: 0.1 to 200 dB.

## Returns

<NRf>

## Examples

DISPLAY:BIbEMISSIONS:Y:SCALE 100 sets the vertical range to 100 dB.

## DISPlay:BIbEmissions:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Bluetooth InBand Emission view.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Display commands

## Syntax

```
DISPlay:BIbEmissions:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:BIbEMISSIONS:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:BIBEmissions:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Bluetooth InBand Emission graph.

### Conditions

Measurement views: Bluetooth InBand Emission

### Group

Display commands

### Syntax

```
DISPlay:BIBEmissions:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:BIBEmissions:Y[:SCALE]:OFFSet?
```

### Related Commands

[DISPlay:BIBEmissions:Y\[:SCALE\]](#)

### Arguments

<value> ::= <NRf> specifies the vertical offset. Range: -170 to +50 dBm.

### Returns

<NRf>

### Examples

DISPLAY:BIBEMISSIONS:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBm.

## DISPlay:BLUetooth:CONStE:TRACe:GRATicule:GRID:STATe

Sets or queries the display state of Constellation graticule grid.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Display commands

### Syntax

```
DISPlay:BLUetooth:CONStE:TRACe:GRATicule:GRID:STATe { 0 | 1 | OFF | ON }
```

```
DISPlay:BLUetooth:CONStE:TRACe:GRATicule:GRID:STATe?
```

### Arguments

ON or 1 turns on the grid.

OFF or 0 turns off the grid.

## Returns

See Arguments.

## Examples

`DISPlay:BLUEtooth:CONSte:TRACe:GRATicule:GRID:STATe 1` turns on the grid in the Constellation plot.

## **DISPlay:BLUEtooth:CONSte:WINDow:TRACe:GRATicule:GRID:STATe**

Sets or queries whether to show or hide the graticule grid on the screen in the Bluetooth Constellation display.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:CONSte:WINDow:TRACe:GRATicule:GRID:STATe STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:BLUEtooth:CONSte:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

`OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

## Returns

See Arguments.

## Examples

`DISPlay:BLUEtooth:CONSte:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the screen.

## **DISPlay:BLUEtooth:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe**

Sets or queries graticule grid hidden or showing state in the Bluetooth eye diagram display.

## Conditions

Measurement views: Bluetooth Eye diagram

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe { 0 | 1 | OFF | ON }
DISPlay:BLUEtooth:EDIagram:WINDow:TRACe:GRATicule:GRID: STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

DISPLAY:BLUETOOTH:EDIAGRAM:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid in the eye diagram.

## DISPlay:BLUEtooth:EDIagram:Y[:SCALE]

Sets or queries the vertical range of the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth Eye diagram

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:EDIagram:Y[:SCALE] <value>
DISPlay:BLUEtooth:EDIagram:Y[:SCALE]?
```

## Arguments

<value> ::= <NRf> the vertical range (1 to 100)

## Returns

<NRf>

## Examples

DISPlay:BLUEtooth:EDIagram:Y:SCALE 2.5 sets the vertical range to 2.5 in the eye diagram.

## DISPlay:BLUEtooth:EDIagram:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform in the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth Eye diagram

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:EDIagram:Y[:SCALE]:AUTO
```

## Arguments

None

## Returns

DISPLAY:BLUETOOTH:EDIAGRAM:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:BLUEtooth:EDIagram:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) of the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth Eye diagram

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:EDIagram:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:BLUEtooth:EDIagram:Y[:SCALE]:OFFSet?
```

## Arguments

<value> ::= <NRf> the minimum vertical scale offset, range (-50 to +50).

## Returns

<NRf>

## Examples

DISPlay:BLUEtooth:EDIagram:Y:SCALE:OFFSet -0.5 sets the vertical offset to -0.5.

## DISPlay:BLUEtooth:FDVTime:WINDow:TRACe:GRATicule:GRID:STATe

Shows or hides the graticule grid in the Bluetooth Frequency Deviation vs. Time display.



## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:FDVTime:WINDow:TRACe:GRATicule:GRID:STATe{OFF | ON | 0 | 1}
```

```
DISPlay:BLUEtooth:FDVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

`DISPlay:BLUEtooth:FDVTime:WINDow:TRACe:GRATicule:GRID:STATe ON` specifies to show the graticule grid.

## DISPlay:BLUEtooth:FDVTime:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform to the screen in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:FDVTime:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISPlay:BLUEtooth:FDVTime:Y:SCALE:AUTO` rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:BLUEtooth:FDVTime:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:FDVTime:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:BLUEtooth:FDVTime:Y[:SCALE]:OFFSet?
```

## Arguments

<value> ::= <NRf> the vertical offset. Range: -60 MHz to +60 MHz.

## Returns

<NRf>

## Examples

DISPlay:BLUEtooth:FDVTime:Y:SCALE:OFFSet -14.5E+3 sets the vertical offset to -14.5 kHz.

## DISPlay:BLUEtooth:MEASview:DElete (No Query Form)

Deletes the specified Bluetooth analysis view.

## Conditions

Measurement views: All.

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:MEASview:DElete
```

```
{CONS|SUMM|FDVT|EDI|BOBW|STAB|BIBE|DRIF}
```

## Related Commands

DISPlay:BLUEtooth:MEASview:NEW, DISPlay:BLUEtooth:MEASview:SElect

## Arguments

CONS deletes the BT Constellation view.

SUMM deletes the BT Summary view.

FDVT deletes the BT Freq Dev vs. Time view. EDI deletes the BT Eye Diagram view.

BOBW deletes the BT 20dB BW view. STAB deletes the BT Symbol Table view. BIBE deletes the BT InBand Emission view. DRIF deletes the BT Drift Table view.

---

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

DISPLAY:BLUETOOTH:MEASVIEW:DELETE CONS deletes the BT Constellation view.

## DISPlay:BLUEtooth:MEASview:NEW (No Query Form)

Displays a new measurement view for Bluetooth measurements.

## Conditions

Measurement views: Bluetooth measurements.

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:MEASview:NEW  
{CONS | SUMM | FDVT | EDI | BOBW | STAB | BIBE | DRIF}
```

## Arguments

CONS creates a new BT Constellation view.

SUMM creates a new BT Summary view.

FDVT creates a new BT Freq Dev vs. Time view.

EDI creates a new BT Eye Diagram view.

BOBW creates a new BT 20dB BW view.

STAB creates a new BT Symbol Table view.

BIBE creates a new BT InBand Emission view.

DRIF creates a new BT Drift Table view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

DISPLAY:BLUETOOTH:MEASVIEW:NEW CONS creates a new Bluetooth Constellation view.

## DISPlay:BLUEtooth:MEASview:SElect

Selects a measurement view on the screen in the Bluetooth measurements. The query command returns the currently selected view.

## Conditions

Measurement views: Bluetooth measurements.

## Group

Display commands

## Syntax

```
DISPlay:BLUEtooth:MEASview:SElect  
{ CONS | SUMM | FDVT | EDI | BOBW | STAB | BIBE | DRIF }  
DISPlay:BLUEtooth:MEASview:SElect?
```

## Arguments

CONS selects the BT Constellation view.

SUMM selects the BT Summary view.

FDVT selects the BT Freq Dev vs. Time view.

EDI selects the BT Eye Diagram view.

BOBW selects the BT 20dB BW view.

STAB selects the BT Symbol Table view.

BIBE selects the BT InBand Emission view.

DRIF selects the BT Drift Table view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Returns

See Arguments.

## Examples

```
DISPLAY:BLUETOOTH:MEASVIEW:SELECT EDI selects the Bluetooth Eye Diagram view.
```

## DISPlay:BOBW:SElected:BANDwidth

Sets or queries the bandwidth to measure in the Bluetooth 20 dB BW view.

## Conditions

Measurement views: Bluetooth 20 dB BW

## Group

Display commands

## Syntax

```
DISPlay:BOBW:SElected:BANDwidth { XDBM | XDB }  
DISPlay:BOBW:SElected:BANDwidth?
```

## Arguments

XDBM selects the x dB bandwidth to measure.

XDB selects the x dB bandwidth to measure.

## Returns

See Arguments.

## Examples

```
DISPlay:BOBW:SElected:BANDwidth XDB selects the x dB bandwidth to measure.
```

## DISPlay:CCDF:LEGend:STATe

Determines whether to show or hide the trace legend in the CCDF view. The legend indicates the trace detection and function on the screen for each displayed CCDF trace.

## Conditions

Measurement views: CCDF

## Group

Display commands

## Syntax

```
DISPlay:CCDF:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:CCDF:LEGend:STATe?
```

## Arguments

OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

## Returns

See Arguments.

## Examples

```
DISPlay:CCDF:LEGend:STATe ON shows the trace legend on the screen.
```

## DISPlay:CCDF:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: CCDF

## Group

Display commands

## Syntax

```
DISPlay:CCDF:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:CCDF:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

DISPlay:CCDF:WINDow:TRACe:GRATicule:GRID:STATe ON shows the graticule grid on the screen in the CCDF view.

## DISPlay:CONSte[:MEASview<y>]:MPHase

Sets or queries the multiplication constant of the phase multiplication constellation display for a CPM signal. This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to CPM.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Display commands

## Syntax

```
DISPlay:CONSte[:MEASview<y>]:MPHase { P1 | P2 | P4 | P8 | P16 | P32 }
```

```
DISPlay:CONSte[:MEASview<y>]:MPHase?
```

## Arguments

The following table shows the arguments and phase multiplier.

**Table 36: Phase multiplication**

Argument	Phase multiplier
P1	1

Table continued...

Argument	Phase multiplier
P2	2
P4	4
P8	8
P16	16
P32	32

## Returns

See Arguments.

## Examples

`DISPlay:CONSte[:MEASview<y>]:MPHase P8` sets the phase multiplier to 8 for measview 2 of constellation.

## DISPlay:CONSte[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

The parameter <x> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Display commands

## Syntax

```
DISPlay:CONSte[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:CONSte[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

`DISPlay:CONSte:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the screen for measview 2 of constellation.

## DISPlay:DDEMod[:MEASview<y>]:DELeTe (No query form)

Deletes the measurement view in the general purpose digital modulation measurements.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58 )

### Conditions

Measurement views: General purpose digital modulation

### Group

Display commands

### Syntax

```
DISPlay:DDEMod[:MEASview<y>]:DELeTe { CONStE | DIQVtime | EDIagram | EVM | FDVTime | MERRor | PERRor | SIGNalqual | STABle | TDIagram }
```

### Arguments

The following table lists the arguments. The arguments are the string type.

**Table 37: Modulation measurement views**

Argument	View
CONStE	Constellation
DIQVtime	Demodulated I&Q versus Time
EDIagram	Eye diagram
EVM	EVM (Error Vector Magnitude) versus Time
FDVTime	Frequency deviation versus Time
MERRor	Magnitude error versus Time
PERRor	Phase error versus Time
SIGNalqual	Signal quality
STABle	Symbol table
TDIagram	Trellis diagram

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

### Examples

DISPlay:DDEMod:MEASview1:DELeTe CONStE deletes the constellation view with measurement view 1.

## DISPlay:DDEMod:MEASview<y>[:SOURce<x>]:NEW (No query form)

Displays a new measurement view in the general purpose digital modulation measurements.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).



## Conditions

Measurement views: General purpose digital modulation

## Group

Display commands

## Syntax

```
DISPlay:DDEMod:MEASview<y>[:SOURce<x>]:NEW { CONSTe | DIQVtime | EDIagram | EVM |
FDVTime | MERRor | PERRor | SIGNALqual | STABLE | TDIagram }
```

## Arguments

<x> = 1 to 8, representing Range of Source.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

DISPlay:DDEMod:MEASview:SOURce1:NEW CONSTe creates the constellation view with measurement view 1 for source1.

## DISPlay:DDEMod[:MEASview<y>]:SELEct

Selects a measurement view in the general purpose digital modulation measurements on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to

```
DISPlay:WINDow:OPTimized:MEASurement? query.
```

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Conditions

Measurement views: General purpose digital modulation

## Group

Display commands

## Syntax

```
DISPlay:DDEMod[:MEASview<y>]:SELEct { CONSTe | DIQVtime | EDIagram | EVM | FDVTime
| MERRor | PERRor | SIGNALqual | STABLE | TDIagram }
```

```
DISPlay:DDEMod[:MEASview<y>]:SELEct?
```

## Arguments

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

`DISPlay:DDEModMEASview1:SElect CONSTe` selects the constellation view with measurement view 1.

## DISPlay:DDEMod[:SOURce<x>]:RADix

Sets or queries the base of symbols of the measurement source. This command is effective in the symbol table.

### Conditions

Measurement views: Symbol table

Source keyword is optional, if not present Source 1 will be default source.

### Group

Display commands

### Syntax

`DISPlay:DDEMod[:SOURce<x>]:RADix { BINary | HEXadecimal }`

`DISPlay:DDEMod[:SOURce<x>]:RADix?`

### Arguments

<x> = 1 to 8, representing Range of Source.

`BINary` selects binary notation.

`HEXadecimal` selects hexadecimal notation.

## Examples

`DISPlay:DDEMod:SOURce1:RADix BINary` selects the binary notation of the measurement source symbol table.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]

Sets or queries the measurement source horizontal scale (full-scale time) for the time measurements in the general purpose digital modulation analysis. Programming a specified scale sets `DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:AUTO:STATE OFF`.

### Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

### Group

Display commands

### Syntax

`DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]<value>`

`DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]?`

## Related Commands

[DISPlay:DDEMod\[:SOURce<x>\]:X\[:SCALe\]:AUTO:STATe](#)

[DISPlay:DDEMod\[:SOURce<x>\]:X\[:SCALe\]:OFFSet](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

<value>::=<NRf> specifies the horizontal scale in full-scale time.

Use the `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MAXimum` and `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MINimum?` queries to get the upper and lower limits of the setting range. The unit can be changed by the `[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs` command.

## Returns

<NRf>

## Examples

`DISPlay:DDEMod:SOURce1:X:SCALe 1.5 µs`, sets the horizontal scale of the measurement source to 1.5 µs.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO (No query form)

Sets the horizontal scale of the measurement source automatically to fit the waveform to the screen in the general purpose digital modulation analysis. Executing this command sets `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO:STATeON`.

## Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

## Group

Display commands

## Syntax

`DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO`

## Related Commands

[DISPlay:DDEMod\[:SOURce<x>\]:X\[:SCALe\]:AUTO:STATe](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

## Examples

`DISPlay:DDEMod:SOURce1:X:SCALe:AUTO` sets the horizontal scale of the measurement source1 automatically to fit the waveform to the screen.

## **DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO:STATe**

Determines whether to set the horizontal scale of the measurement source automatically or manually.

### **Conditions**

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

### **Group**

Display commands

### **Syntax**

```
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:AUTO:STATe?
```

### **Arguments**

<x> = 1 to 8, representing Range of Source.

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]` and `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet` commands.

ON or 1 specifies that the horizontal scale is set automatically.

### **Returns**

ON or 1.

OFF or 0.

### **Examples**

`DISPlay:DDEMod:SOURce1:X:SCALe:AUTO:STATeON` specifies that the horizontal scale of the measurement source1 is set automatically.

## **DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MAXimum? (Query only)**

Queries the upper limit of the measurement source horizontal scale setting range.

### **Conditions**

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

### **Group**

Display commands

### **Syntax**

```
DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:MAXimum?
```

## Returns

<NRf> The upper limit of the horizontal scale setting range.

The unit can be changed by the [SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs command.

## Examples

DISPlay:DDEMod:SOURce1:X:SCALE:MAXimum? might return 18.135E-3, indicating that the upper limit of the measurement source1 horizontal scale setting range is 18.135 ms.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:MINimum? (Query only)

Queries the lower limit of the measurement source horizontal scale setting range.

## Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

## Group

Display commands

## Syntax

DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:MINimum?

## Returns

<NRf> The lower limit of the horizontal scale setting range.

The unit can be changed by the [SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs command.

## Examples

DISPlay:DDEMod:SOURce1:X:SCALE:MINimum? might return 10.0E-9, indicating that the lower limit of the measurement source1 horizontal scale setting range is 10.0 ns.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value of the measurement source (left edge) for the time measurements in the general purpose digital modulation analysis.

Programming a specified offset sets DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:AUTO:STATE OFF.

## Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

## Group

Display commands

## Syntax

DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet<value>

DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet?

## Related Commands

[DISPlay:DDEMod\[:SOURce<x>\]:X\[:SCALe\]:AUTO:STATe](#)

[DISPlay:DDEMod\[:SOURce<x>\]:X\[:SCALe\]](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

<value> ::= <NRf> specifies the minimum horizontal value.

Use the `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MAXimum?` and `DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MINimum?` queries to get the upper and lower limits of the setting range.

The unit can be changed by the `[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs` command.

## Returns

<NRf>

## Examples

`DISPlay:DDEMod:SOURce1:X:SCALe:OFFSet 20.075 µs`, sets the minimum horizontal value of the measurement source to 20.075 µs.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MAXimum? (Query only)

Queries the measurement source upper limit of the horizontal offset setting range.

## Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

## Group

Display commands

## Syntax

DISPlay:DDEMod[:SOURce<x>]:X[:SCALe]:OFFSet:MAXimum?

## Returns

<NRf> The upper limit of the horizontal offset setting range. The unit can be changed by the `[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs` command.

## Examples

DISPlay:DDEMod:SOURce1:X:SCALE:OFFSet:MAXimum? might return -1.812E-3, indicating that the measurement source 1 upper limit of the horizontal offset setting range is -1.812 ms.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:OFFSet:MINimum? (Query only)

Queries the measurement source lower limit of the horizontal offset setting range.

### Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

### Group

Display commands

### Syntax

DISPlay:DDEMod:X[:SCALE]:OFFSet:MINimum?

### Returns

<NRf> The lower limit of the horizontal offset setting range. The unit can be changed by the [SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs command.

## Examples

DISPlay:DDEMod:SOURce1:X:SCALE:OFFSet:MINimum? might return -16.28E-3, indicating that the measurement source1 lower limit of the horizontal offset setting range is -16.28 ms.

## DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:RESet (No query form)

Presets the horizontal scale of the measurement source to the default value for the time measurements in the general purpose digital modulation analysis.

### Conditions

Measurement views: EVM versus Time, Magnitude error versus Time, Phase error versus Time

Source keyword is optional, if not present Source1 will be default source.

### Group

Display commands

### Syntax

DISPlay:DDEMod[:SOURce<x>]:X[:SCALE]:RESet

### Arguments

<x> = 1 to 8, representing Range of Source.

## Examples

`DISPlay:DDEMod:SOURCE1:X:SCALE:RESet` presets the horizontal scale of the measurement source1 to the default value.

## DISPlay:DIAGram:X[:SCALE]

Sets or queries the horizontal range for the eye and trellis diagrams in the general purpose digital modulation analysis.

### Conditions

Measurement views: Eye diagram, Trellis diagram

### Group

Display commands

### Syntax

`DISPlay:DIAGram:X[:SCALE] <value>`

`DISPlay:DIAGram:X[:SCALE]?`

### Arguments

`<value>::=<NR1>` specifies the horizontal range.

Range: 1 to 16 symbols.

### Returns

`<NR1>`

## Examples

`DISPLAY:DIAGRAM:X:SCALE 3` sets the horizontal range to 3 symbols in the eye and trellis diagrams.

## DISPlay:DIAGram:X[:SCALE]:RESet (No Query Form)

Presets the horizontal scale to the default value for the eye and trellis diagrams in the general purpose digital modulation analysis.

### Conditions

Measurement views: Eye diagram, Trellis diagram

### Group

Display commands

### Syntax

`DISPlay:DIAGram:X[:SCALE]:RESet`

### Arguments

None



## Examples

`DISPLAY:DIAGRAM:X:SCALE:RESET` presets the horizontal scale to the default value for the eye and trellis diagrams.

## DISPlay:DIQVtime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Display commands

### Syntax

```
DISPlay:DIQVtime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:DIQVtime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

### Returns

See Arguments.

## Examples

`DISPlay:DIQVtime:MEASview1:WINDow:TRACe:GRATicule:GRID:STATeON` shows the graticule grid on the screen in the Demodulated I&Q versus Time view 1.

## DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]

Sets or queries the vertical range of the Demodulated I&Q versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Display commands

## Syntax

```
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe] <value>
```

```
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]?
```

## Arguments

<value>::=<NRf> specifies the vertical range. Range: 1  $\mu$ V to 10 V.

## Returns

<NRf>

## Examples

DISPlay:DIQVtime1:MEASview1:Y:SCALe 2.5 sets the vertical range to 2.5 V in the Demodulated I&Q versus Time measurement view 1.

## DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the Demodulated I & Q versus Time display.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Display commands

## Syntax

```
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALe]:AUTO
```

## Arguments

None

## Examples

DISPlay:DIQVtime:MEASview1:Y:SCALe:AUTO sets the vertical scale automatically to fit the waveform to the screen measurement view 1.

## DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) of the Demodulated I&Q versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demodulated I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Display commands

### Syntax

```
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:DIQVtime[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

### Arguments

<value> ::= <NRf> specifies the vertical offset. Range: -5 to +5 V.

### Returns

<NRf>

### Examples

DISPlay:DIQVtime:MEASview1:Y:SCALE:OFFSet -0.5 sets the vertical offset to -0.5 V in the Demodulated I&Q versus Time measurement view 1.

## DISPlay:DPX:DGRam:TIME[:SCALE]:OFFSet:DIVision

Sets or queries the DPXogram vertical time offset in divisions.

### Conditions

Measurement views: DPX spectrum

### Group

Display commands

### Syntax

```
DISPlay:DPX:DGRam:TIME[:SCALE]:OFFSet:DIVision <value>
```

```
DISPlay:DPX:DGRam:TIME[:SCALE]:OFFSet:DIVision?
```

### Arguments

<value> ::= <NRf> vertical offset value in divisions.

## Returns

<NRf>

## Examples

DISPLAY:DPX:DGRAM:TIME:SCALE:OFFSET:DIVISION 10 sets the vertical time off set to 10 divisions.

## DISPlay:DPX:DGRam:TIME[:SCALe]:PDIVision

Sets or queries the DPXogram vertical time scale per division.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

DISPlay:DPX:DGRam:TIME[:SCALe]:PDIVision <value>

DISPlay:DPX:DGRam:TIME[:SCALe]:PDIVision?

## Arguments

<value>::<NRf> vertical value in divisions.

## Returns

<NRf>

## Examples

DISPLAY:DPX:DGRAM:TIME:SCALE:PDIVISION 10 sets the vertical time scale to 10 seconds per division.

## DISPlay:DPX:DGRam:TIME[:SCALe]:RESet (No Query Form)

Presets the time scale and offset to the default value for the DPXogram measurement.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

DISPlay:DPX:DGRam:TIME[:SCALe]:RESet

## Arguments

None

## Examples

`DISPLAY:DPX:DGRAM:TIME:SCALE:RESET` presets the time scale and offset to the default value.

## DISPlay:DPX:DGRam:TSTamp:STATe

Determines whether to display the time stamp readout in the DPXogram display.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

```
DISPlay:DPX:DGRam:TSTamp:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:DPX:DGRam:TSTamp:STATe?
```

## Arguments

`OFF` or `0` hides the time stamp readout.

`ON` or `1` shows the time stamp readout.

## Returns

See Arguments.

## Examples

`DISPLAY:DPX:DGRAM:TSTAMP:STATE 0` hides the time stamp readout in the DPXogram display.

## DISPlay:DPX:DGRam:Y[:SCALE]:AUTO (No Query Form)

Rescales the height axis automatically to fit the waveform to the screen in the DPXogram display.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

```
DISPlay:DPX:DGRam:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISPLAY:DPX:DGRAM:Y:AUTO` rescales the height scale automatically to fit the waveform to the DPXogram display.

## DISPlay:DPX:DGRam:Y[:SCALe]:RESet (No Query Form)

Resets the height scale of the DPXogram display to the default values: Height position = 0 dB and Height scale = 100 dB.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

`DISPlay:DPX:DGRam:Y[:SCALe]:RESet`

## Arguments

None

## Examples

`DISPLAY:DPX:DGRAM:Y:RESET` resets the height scale of the DPXogram display.

## DISPlay:DPX:LEGend:STATe

Determines whether to show or hide the trace legend on the display. The legend indicates the trace detection and function on the screen for each displayed trace.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

`DISPlay:DPX:LEGend:STATe { OFF | ON | 0 | 1 }`

`DISPlay:DPX:LEGend:STATe?`

## Arguments

OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

## Returns

See Arguments.

## Examples

`DISPLAY:DPX:LEGEND:STATE ON` shows the trace legend on the screen.

## DISPlay:DPX:PHASe:Y[:SCALe]:AXIS

Sets or queries the vertical axis representation.

## Conditions

Measurement views: DPX Phase

## Group

Display commands

## Syntax

```
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS { MODUlopi | CONTInuous }
```

```
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS?
```

## Arguments

`MODUlopi` (modulo  $\pi$ ) shows the phase constrained within  $\pm 180^\circ$  along the vertical axis.

`CONTInuous` shows the phase as continuous quantity along the vertical axis.

## Returns

See Arguments.

## Examples

`DISPLAY:DPX:PHASe:Y[:SCALE]:AXIS MODUlopi` selects modulo  $\pi$  representation for the vertical axis.

## DISPlay:DPX:PHASe:Y[:SCALe]:AXIS:OFFSet

Sets or queries the vertical offset in the DPX Phase view.

## Conditions

Measurement views: DPX Phase

## Group

Display commands

## Syntax

```
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS:OFFSet <value>
```

```
DISPlay:DPX:PHASe:Y[:SCALe]:AXIS:OFFSet?
```

## Arguments

<value> : : <NRf> specifies the vertical offset. Range:  $-180^\circ$  to  $+180^\circ$ .

## Returns

<NRf>

## Examples

DISPLAY:DPX:PHASE:Y[:SCALE]:AXIS:OFFSET -158.5 sets the vertical offset to  $-158.5^\circ$  in the view.

## DISPlay:DPX:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

```
DISPlay:DPX:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:DPX:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

DISPLAY:DPX:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen in the DPX spectrum view.

## DISPlay:DPX:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the DPX spectrum view.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands



## Syntax

DISPlay:DPX:Y[:SCALe]:OFFSet <value> DISPlay:DPX:Y[:SCALe]:OFFSet?

## Arguments

<value>::<NRf> specifies the vertical offset. Range: -270 to +150 dBm.

The amplitude unit is set by the [SENSe]:POWer:UNITs command (default: dBm).

You can omit the unit in the argument. When you include the unit in the argument, only dBm is allowed. For the other amplitude units, omit the unit in the argument.

## Returns

<NRf>

## Examples

DISPlay:DPX:Y[:SCALe]:OFFSet -12.5dBm sets the vertical offset to -12.5 dBm.

## DISPlay:DPX:Y[:SCALe]:PDIVision

Sets or queries the vertical scale (per division) in the DPX spectrum view.

## Conditions

Measurement views: DPX spectrum

## Group

Display commands

## Syntax

DISPlay:DPX:Y[:SCALe]:PDIVision <value>

DISPlay:DPX:Y[:SCALe]:PDIVision?

## Related Commands

[\[SENSe\]:POWer:UNITs](#)

## Arguments

<value>::<NRf> specifies the vertical scale (per division).

Range: 2 to 20 dB/div.

## Returns

<NRf>

## Examples

DISP:DPX:Y:SCAL:PDIV 0.5 sets the vertical scale to 0.5 dB/div.

## **DISPlay:EDiagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen in the eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

### **Group**

Display commands

### **Syntax**

```
DISPlay:EDiagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:EDiagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

### **Arguments**

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

### **Returns**

See Arguments.

### **Examples**

```
DISPlay:EDiagram:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON shows the graticule grid on the screen in the eye diagram for measview2.
```

## **DISPlay:EDiagram[:MEASview<y>]:Y[:SCALe]**

Sets or queries the vertical range of the Eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

### **Group**

Display commands

## Syntax

```
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]<value>
```

```
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]?
```

## Arguments

<value>::<NRf> specifies the vertical range. Range: 1  $\mu$  to 100 (unitless).

## Returns

<NRf>

## Examples

DISPlay:EDIagram:MEASview2:Y:SCALE 2.5 sets the vertical range to 2.5 in the eye diagram for measview2.

## DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Display commands

## Syntax

```
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPlay:EDIagram:MEASview2:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen for measview2 of the Eye diagram.

## DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) of the eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Display commands

## Syntax

```
DISPlay:EDIagram[:MEASview<y>]:Y[:SCALE]:OFFSet<value>
```

```
DISPlay:EDIagram[:MEASview<y>]Y[:SCALE]:OFFSet?
```

## Arguments

<value>: :<Nrf> specifies the minimum vertical value.

Range: -50 to +50 (unitless).

## Returns

<Nrf>

## Examples

DISPlay:EDIagram:MEASview2:Y:SCALE:OFFSet -0.5 sets the vertical offset to -0.5 in the eye diagram measview2.

## DISPlay:EMC:INCLude:MARGin

Sets or queries to display or hide the Margin by enabling or disabling the check box in the Prefs tab of the EMC Settings control panel.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:INCLude:MARGin { 0 | OFF | 1 | ON }
```

```
DISPlay:EMC:INCLude:MARGin?
```

## Arguments

0 or OFF hides the Margin

1 or ON displays the margin

## Returns

<Boolean> 0, 1

## Examples

`DISPLAY:EMC:INCLUDE:MARGIN` returns 1, indicating that the Margin is displayed.

## DISPlay:EMC:MARKer:SHOW:STATe

Sets or queries to show or hide the marker readout in the EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

### Syntax

```
DISPlay:EMC:MARKer:SHOW:STATe { 0 | 1 | ON | OFF }
```

```
DISPlay:EMC:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the marker readout

1 or ON shows the marker readout

### Returns

<Boolean> 0, 1

## Examples

`DISPLAY:EMC:MARKER:SHOW:STATE ON` shows the readout for the marker on the display.

## DISPlay:EMC:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale to the default values in the EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

### Syntax

```
DISPlay:EMC:RESet:SCALE
```

### Arguments

None

## Returns

None

## Examples

`DISPLAY:EMC:RESET:SCALE` resets the horizontal and vertical scale to the default values.

## DISPlay:EMC:SCALE:LOG:STATe

Determines whether or not to set the horizontal axis logarithmic in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

`DISPlay:EMC:SCALE:LOG:STATe { 0 | OFF | 1 | ON }`

`DISPlay:EMC:SCALE:LOG:STATe?`

## Arguments

OFF or 0 sets the horizontal axis linear (default).

ON or 1 sets the horizontal axis logarithmic.

## Returns

<Boolean> 0, 1

## Examples

`DISPLAY:EMC:SCALE:LOG:STATE ON` sets the horizontal axis logarithmic in the EMC view.

## DISPlay:EMC:SElect:SPOT:NUMBER

Sets or queries the status of current spot number in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

`DISPlay:EMC:SElect:SPOT:NUMBER <value>`

`DISPlay:EMC:SElect:SPOT:NUMBER?`

## Arguments

<value> = <NR1>

## Returns

<NR1>

## Examples

DISPLAY:EMC:SELECT:SPOT:NUMBER might return +2, indicating that spot 2 is made current spot and it is enabled. (+ represents enabling and – represents disabling.)

## DISPlay:EMC:SHOW:LIMit

Sets or queries the displayed limit format in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:SHOW:LIMit {LINE|SHAD|OFF}
```

```
DISPlay:EMC:SHOW:LIMit?
```

## Arguments

LINE – Sets the limit in Line format

SHAD – Sets the limit in Shaded format

OFF – Hides the Limit

## Returns

<String> Either LINE or SHAD or OFF

## Examples

DISPLAY:EMC:SHOW:LIMIT might return LINE, indicating that the limit is displayed in line format.

## DISPlay:EMC:SHOW:LIMit:LEGend

Sets or queries to display or hide the limit legends and label on the plot by enabling or disabling the check box in the Prefs tab of the EMC Settings control panel.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:SHOW:LIMit:LEGend { 0 | OFF | 1 | ON }
```

```
DISPlay:EMC:SHOW:LIMit:LEGend?
```

## Arguments

OFF or 0 hides the limit legends and label.

ON or 1 display the limit legends and label.

## Returns

<Boolean> 0,1

## Examples

DISPLAY:EMC:SHOW:LIMIT:LEGEND might return 0, indicating that the limit legends and label are hidden on the plot.

## DISPlay:EMC:SHOW:SPOT:INDIcator

Sets or queries to display or hide the spots on the plot in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:SHOW:SPOT:INDIcator {0 | OFF | 1 | ON }
```

```
DISPlay:EMC:SHOW:SPOT:INDIcator?
```

## Arguments

OFF or 0 hides the spots

ON or 1 display the spots

## Returns

<Boolean> 0,1

## Examples

DISPLAY:EMC:SHOW:SPOT:INDICATOR1 displays the spots on the plot in EMC View.



## DISPlay:EMC:SHOW:WIZArD

Sets or queries to display or hide the EMC wizard when EMCVu is launched by enabling or disabling the check box in the Prefs tab of the EMC Settings control panel.

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

### Syntax

```
DISPlay:EMC:SHOW:WIZArD {0 | OFF | 1 | ON}
```

```
DISPlay:EMC:SHOW:WIZArD?
```

### Arguments

OFF or 0 hides the EMC wizard

ON or 1 displays the EMC wizard

### Returns

<Boolean> 0,1

### Examples

DISPLAY:EMC:SHOW:WIZARD ON displays the EMC wizard when EMCVu is launched.

## DISPlay:EMC:SPOT<x>:SElect

Sets or queries to enable or disable the selected spot in emission results of EMC view.

<x> = <NR1> Indicates the spot number

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

### Syntax

```
DISPlay:EMC:SPOT<x>:SElect { 0 | 1 | ON | OFF }
```

```
DISPlay:EMC:SPOT<x>:SElect ?
```

### Arguments

OFF or 0 disables the spot

1 or ON enables the spot

## Returns

<Boolean> 0,1

## Examples

DISPlay:EMC:SPOT:SELEct? might return 1, indicating that spot 2 is enabled.

## DISPlay:EMC:TRACe:LEGEnd

Sets or queries to display or hide the trace legend by enabling or disabling the check box in the Prefs tab of the EMC Settings control panel.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:TRACe:LEGEnd { 0 | OFF | 1 | ON }
```

```
DISPlay:EMC:TRACe:LEGEnd?
```

## Arguments

0 or OFF hides the Trace Legend

1 or ON display the Trace Legend

## Returns

<Boolean> 0, 1

## Examples

DISPLAY:EMC:TRACE:LEGEND returns 0, indicating that the trace legend is hidden.

## DISPlay:EMC:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries to display or hide the Graticule grid in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:WINDow:TRACe:GRATicule:GRID:STATe { 0 | 1 | ON | OFF }
```

```
DISPlay:EMC:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the Graticule grid in the graph 1 or

ON display the Graticule grid in the graph

### Returns

<Boolean> 0, 1

### Examples

```
DISPLAY:EMC:WINDOW:TRACE:GRATICULE:GRID:STATE OFF hides the graticule grid in the EMC view.
```

## DISPlay:EMC:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the complete screen in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

### Syntax

```
DISPlay:EMC:X[:SCALE]:AUTO
```

### Arguments

None

### Returns

None

### Examples

```
DISPLAY:EMC:X[:SCALE]:AUTO sets the horizontal scale automatically to fit the waveform to the complete screen.
```

## DISPlay:EMC:X[:SCALE]:START

Sets and queries the Zoom Start frequency in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Display commands

## Syntax

DISPlay:EMC:X[:SCALE]:START <value>

DISPlay:EMC:X[:SCALE]:START?

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

DISPLAY:EMC:X[:SCALE]:START might return 64.29, indicating that the zoom start frequency is 64.2 kHz.

## DISPlay:EMC:X[:SCALE]:STOP

Sets and queries the Zoom stop frequency in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

DISPlay:EMC:X[:SCALE]:STOP <value>

DISPlay:EMC:X[:SCALE]:STOP?

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

DISPLAY:EMC:X[:SCALE]:STOP might return 732.29, indicating that the zoom stop frequency is 732.2 kHz.

## DISPlay:EMC:Y[:SCALE]

Sets or queries the vertical scale value in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:Y[:SCALE] <value>
```

```
DISPlay:EMC:Y[:SCALE]?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

DISPLAY:EMC:Y[:SCALE] 100 sets the vertical scale to 100 dB.

## DISPlay:EMC:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the complete screen in the EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:Y[:SCALE]:AUTO
```

## Arguments

None

## Returns

None

## Examples

DISPLAY:EMC:Y[:SCALE]:AUTO sets the vertical scale automatically to fit the waveform to the complete screen.

## DISPlay:EMC:Y[:SCALE]:OFFSet

Sets or queries the vertical Position (offset) value in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Display commands

## Syntax

```
DISPlay:EMC:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:EMC:Y[:SCALE]:OFFSet?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

DISPLAY:EMC:Y[:SCALE]:OFFSET might return 45.74, indicating the vertical Position (offset) value is 45.7 dBuV.

## DISPlay:EVM[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule grid view state for the EVM view.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Display commands

## Syntax

```
DISPlay:EVM:MEASview<y>:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:EVM:MEASview<y>:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

OFF or 0

ON or 1

## Examples

`DISPlay:EVM:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the EVM Measview 2.

## DISPlay:EVM[:MEASview<y>]:Y[:SCALE]

Sets or queries the vertical range of the EVM versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Display commands

## Syntax

`DISPlay:EVM[:MEASview<y>]:Y[:SCALE]<value>`

`DISPlay:EVM[:MEASview<y>]:Y[:SCALE]?`

## Related Commands

[\*DISPlay:EVM:\[MEASview1\]:Y\[:SCALE\]:OFFSet\*](#)

## Arguments

<value>::<NRf> specifies the vertical range.

Range: 1 to 100%.

## Returns

<NRf>

## Examples

`DISPlay:EVM:MEASview1:Y:SCALE 50` sets the vertical range to 50% in the EVM versus Time display Measurement view 1.

## DISPlay:EVM[:MEASview<y>]:Y[:SCALE]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the EVM versus Time display.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Display commands

## Syntax

```
DISPlay:EVM[:MEASview<y>]:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISPlay:EVM:MEASview2:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen of EVM versus Time Measurement view 2.

## DISPlay:EVM[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the EVM versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Display commands

## Syntax

```
DISPlay:EVM[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:EVM[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:EVM\[:MEASview<y>\]:Y\[:SCALE\]](#)

## Arguments

<value>::<Nrf> specifies the minimum vertical value.

Range: -100 to 100%.



## Returns

<NRf>

## Examples

`DISPlay:EVM:MEASview2:Y:SCALE:OFFSet -9.5` sets the minimum vertical value to -9.5% in the EVM versus Time graph Measurement View 2.

## **DISPlay:FDVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Sets or queries the graticule grid view state for the FDV view.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Display commands

## Syntax

`DISPlay:FDVTime:MEASview<y>:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }`

`DISPlay:FDVTime:MEASview<y>:WINDow:TRACe:GRATicule:GRID:STATe?`

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

OFF

ON

## Examples

`DISPlay:FDVTime:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the Frequency deviation versus Time Measview 2.

## **DISPlay:FDVTime[:MEASview<y>]:Y[:SCALE]**

Sets or queries the vertical range of the Frequency deviation versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Display commands

## Syntax

```
DISPlay:FDVTime:[MEASview<y>]:Y[:SCALE]<value>
```

```
DISPlay:FDVTime:[MEASview<y>]:Y[:SCALE]?
```

## Related Commands

[DISPlay:FDVTime\[:MEASview<y>\]:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::<NRf> specifies the vertical range.

Range: 10 Hz to 120 MHz.

## Returns

<NRf>

## Examples

DISPlay:FDVTime:MEASview1:Y:SCALE 30 MHz sets the vertical range to 30 MHz in the Frequency deviation versus Time display Measurement view 1.

## DISPlay:FDVTime[:MEASview<y>]:Y[:SCALE]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the Frequency deviation versus Time display.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Display commands

## Syntax

```
DISPlay:FDVTime:MEASview<y>:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISPlay:FDVTime:MEASview2:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen of Frequency deviation versus Time Measurement view 2.

## DISPlay:FDVTime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the Frequency deviation versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Display commands

## Syntax

`DISPlay:FDVTime[:MEASview<y>]:Y[:SCALE]:OFFSet<value>`

`DISPlay:FDVTime[:MEASview<y>]:Y[:SCALE]:OFFSet?`

## Related Commands

[DISPlay:FDVTime\[:MEASview<y>\]:Y\[:SCALE\]](#)

## Arguments

<value>::<NRf> specifies the vertical offset.

Range: -60 MHz to +60 MHz.

## Returns

<NRf>

## Examples

`DISPlay:FDVTime:MEASview2:Y:SCALE:OFFSet -14.5 MHz` sets the minimum vertical value to -14.5 MHz in the Frequency deviation versus Time graph Measurement View 2.

## DISPlay:{FSETtling|PSETtling}:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Frequency or Phase settling views.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:MARKer:SHOW:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:{FSETtling|PSETtling}:MARKer:SHOW:STATE?
```

## Arguments

OFF or 0 hides the readout for the selected marker in the view.

ON or 1 shows the readout for the selected marker in the view.

## Returns

See Arguments.

## Examples

```
DISPLAY:FSETTLING:MARKER:SHOW:STATE ON shows the readout for the selected marker in the view.
```

## DISPlay:{FSETtling|PSETtling}:TIME:DECimal

Sets or queries the number of values to the right of the decimal point to include in the Settling Time and Settling Time from Trigger results.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:TIME:DECimal <value>
```

```
DISPlay:{FSETtling|PSETtling}:TIME:DECimal?
```

## Arguments

<value>::<NR1> specifies the number digits to the right of the decimal point to include in Settling Time and Settling Time from Trigger results. Range: 0 to 6.

## Returns

<NR1>

## Examples

`DISPLAY:PSETTLING:TIME:DECIMAL 3` sets the number of values to the right of the decimal point to 3 for the Phase Settling time measurement.

## **DISPlay:{FSETtling|PSETtling}:WINDow:TRACe:GRATICule:GRID:STATE**

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Display commands

### Syntax

```
DISPlay:{FSETtling|PSETtling}:WINDow:TRACe:GRATICule:GRID: STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:{FSETtling|PSETtling}:WINDow:TRACe:GRATICule:GRID: STATE?
```

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

### Returns

See Arguments.

## Examples

`DISPLAY:FSETTLING:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen for the Frequency Settling display.

## **DISPlay:{FSETtling|PSETtling}:X[:SCALE]**

Sets or queries the horizontal scale (full-scale time) of the Settling Time graph. Programming a specified scale sets `DISPlay:{FSETtling|PSETtling}:X[:SCALE] AUTO:STATE OFF`.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

```
DISPlay:{FSETtling|PSETtling}:X[:SCALE] <value>
```

`DISPlay:{FSETtling|PSETtling}:X[:SCALE]?`

## Related Commands

[`DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]:AUTO:STATeDISPlay:`](#)

[`{FSETtling|PSETtling}:X\[:SCALE\]:OFFSet,`](#)

## Arguments

`<value>::={ <Nrf> | MAXimum | MINimum }` specifies the horizontal scale in full-scale time. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [`DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]:MAXimum?`](#) and [`DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]:MINimum?`](#) queries to get the upper and lower limit values of the setting range.

## Returns

See Arguments.

## Examples

`DISPLAY:FSETTLING:X:SCALE 25.6us` sets the horizontal scale to 25.6  $\mu$ s.

## DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Frequency and Phase Settling views. Executing this command sets `DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO:STATe ON`.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

`DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO`

## Related Commands

[`DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]:AUTO:STATe`](#)

## Arguments

None

## Examples

`DISPLAY:FSETTLING:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:{FSETtling|PSETtling}:X[:SCALE]:AUTO:STATE?
```

## Arguments

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the

[DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]](#) and [DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]:OFFSet](#) commands.

ON or 1 specifies that the horizontal scale is set automatically.

## Returns

See Arguments.

## Examples

```
DISPLAY:FSETTLING:X:SCALE:AUTO:STATE ON specifies that the horizontal scale is set automatically.
```

## DISPlay:{FSETtling|PSETtling}:X[:SCALE]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:X[:SCALE]:MAXimum?
```

## Related Commands

[DISPlay:{FSETtling|PSETtling}:X\[:SCALE\]](#)

## Returns

<NRf> The upper limit of the horizontal scale setting range.

## Examples

`DISPLAY:FSETTLING:X:SCALE:MAXIMUM?` might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

## **DISPlay:{FSETtling|PSETtling}:X[:SCALe]:MINimum? (Query Only)**

Queries the lower limit of the horizontal scale setting range.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Display commands

### Syntax

`DISPlay:{FSETtling|PSETtling}:X[:SCALe]:MINimum?`

### Related Commands

[\*DISPlay:FVTime:X\[:SCALe\]\*](#)

### Returns

<NRf> The lower limit of the horizontal scale setting range.

## Examples

`DISPLAY:FSETTLING:X:SCALE:MINIMUM?` might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

## **DISPlay:{FSETtling|PSETtling}:X[:SCALe]:OFFSet**

Sets or queries the minimum horizontal value (left edge) of the Frequency versus Time graph. Programming a specified offset sets `DISPlay:{FSETtling|PSETtling}:X[:SCALe] AUTO:STATE OFF`.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Display commands

### Syntax

`DISPlay:{FSETtling|PSETtling}:X[:SCALe]:OFFSet <value>`

`DISPlay:{FSETtling|PSETtling}:X[:SCALe]:OFFSet?`

### Related Commands

[\*DISPlay:{FSETtling|PSETtling}:X\[:SCALe\]:AUTO:STATE\*](#), [\*DISPlay:{FSETtling|PSETtling}:X\[:SCALe\]\*](#)



## Arguments

`<value>::={ <NRf> | MAXimum | MINimum }` specifies the horizontal offset. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the `DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet:MAXimum?` and `DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet:MINimum?` queries to get the upper and lower limit values of the setting range.

## Returns

See Arguments.

## Examples

`DISPLAY:FSETTLING:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Frequency Settling graph.

## DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

`DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet:MAXimum?`

## Related Commands

`DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet`

## Returns

`<NRf>` The upper limit of the horizontal offset setting range.

## Examples

`DISPLAY:FSETTLING:X:SCALE:OFFSET:MAXIMUM?` might return `338.186000E-6`, indicating that the upper limit of the horizontal offset setting range is 338.186  $\mu$ s.

## DISPlay:{FSETtling|PSETtling}:X[:SCALE]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

DISPlay:{FSETtling|PSETtling}:X[:SCALe]:OFFSet:MINimum?

## Related Commands

*DISPlay:FVTime:X[:SCALe]:OFFSet*

## Returns

<NRf> The lower limit of the horizontal offset setting range.

## Examples

DISPLAY:FSETTLING:X:SCALE:OFFSET:MINIMUM? might return  $-338.186000E-6$ , indicating that the lower limit of the horizontal offset setting range is  $-338.186 \mu\text{s}$ .

## DISPlay:{FSETtling|PSETtling}:Y[:SCALe]

Sets or queries the vertical range of the Frequency versus Time graph.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

DISPlay:{FSETtling|PSETtling}:Y[:SCALe] <value>

DISPlay:{FSETtling|PSETtling}:Y[:SCALe]?

## Related Commands

*DISPlay:{FSETtling|PSETtling}:Y[:SCALe]:OFFSet*

## Arguments

<value>::<NRf> specifies the vertical range. Range: 10 Hz to 120 MHz.

## Returns

<NRf>

## Examples

DISPlay:FSETTLING:Y:SCALE 30E6 sets the vertical range to 30 MHz in the Frequency Settling graph.

## DISPlay:{FSETtling|PSETtling}:Y[:SCALe]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Frequency and Phase Settling views.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISPLAY:FSETTLING:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the Frequency and Phase Settling graphs.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Display commands

## Syntax

```
DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:OFFSet?
```

## Related Commands

[\*DISPlay:{FSETtling|PSETtling}:Y\[:SCALE\]\*](#)

## Arguments

`<value>::<NRf>` specifies the vertical offset. Range: -60 MHz to +60 MHz.

## Returns

`<NRf>`

## Examples

`DISPLAY:FVTIME:Y:SCALE:OFFSET -14.5E6` sets the vertical offset to -14.5 MHz in the Frequency Settling graph.

## DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:PDIVision

Sets or queries the vertical scale (per division) of the Frequency and Phase Settling graphs.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Display commands

### Syntax

```
DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:PDIVision <value>
```

```
DISPlay:{FSETtling|PSETtling}:Y[:SCALE]:PDIVision?
```

### Arguments

<value>::<NRf> specifies the vertical scale (per division).

Range:

- Frequency Settling (Hz/div) : 500 to the Maximum bandwidth of instrument
- Phase Settling (degrees/div): 0.10 to X.

### Returns

<NRf>

### Examples

DISPLAY:PSETTLING:Y:SCALE:PDIVISION 2.7 sets the vertical scale to 27 degrees.

## DISPlay:FVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

```
DISPlay:FVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:FVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

See Arguments.

## Examples

DISPLAY:FVTIME:MEASview1:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the Frequency versus Time view.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]

Sets or queries the horizontal scale (full-scale time) of the Frequency versus Time graph. Programming a specified scale sets DISPlay:FVTime[:MEASview<y>]:X[:SCALE] AUTO:STATE OFF.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

DISPlay:FVTime[:MEASview<y>]:X[:SCALE] <value>

DISPlay:FVTime[:MEASview<y>]:X[:SCALE]?

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATE](#), [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <NRf> | MAXimum | MINimum } specifies the horizontal scale in full-scale time. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:MAXimum?](#) and [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:MINimum?](#) queries to get the upper and lower limit values of the setting range.

## Returns

See Arguments.

## Examples

DISPLAY:FVTIME:MEASview1:X:SCALE 25.6us sets the horizontal scale to 25.6  $\mu$ s.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Frequency versus Time view. Executing this command sets `DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE ON`.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO
```

### Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATE](#)

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Examples

`DISPLAY:FVTIME:MEASview1:X:SCALE:AUTO` sets the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE

Determines whether to set the horizontal scale automatically or manually.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]](#) and [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#) commands.

ON or 1 specifies that the horizontal scale is set automatically.

## Returns

See Arguments.

## Examples

`DISPLAY:FVTIME:MEASview1:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

`DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:MAXimum?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]](#)

## Returns

<NRf> The upper limit of the horizontal scale setting range.

## Examples

`DISPLAY:FVTIME:MEASview1:X:SCALE:MAXIMUM?` might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]](#)

## Returns

<NRf> The lower limit of the horizontal scale setting range.

## Examples

DISPLAY:FVTIME:MEASview1:X:SCALE:MINIMUM? might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Frequency versus Time graph. Programming a specified offset sets  
DISPlay:FVTime[:MEASview<y>]:X[:SCALE] AUTO:STATE OFF.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet <value>
```

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATE](#), [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <NRf> | MAXimum | MINimum } specifies the horizontal offset. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet:MAXimum?](#) and [DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet:MINimum?](#) queries to get the upper and lower limit values of the setting range.



## Returns

See Arguments.

## Examples

`DISPLAY:FVTIME:MEASview1:X:SCALE:OFFSET 800ns` sets the minimum horizontal value to 800 ns in the Frequency versus Time graph.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

`DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Related Commands

`DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet`

## Returns

<NRf> The upper limit of the horizontal offset setting range.

## Examples

`DISPLAY:FVTIME:MEASview1:X:SCALE:OFFSET:MAXIMUM?` might return `-1.812E-3`, indicating that the upper limit of the horizontal offset setting range is `-1.812 ms`.

## DISPlay:FVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

```
DISPlay:FVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet](#)

## Returns

<NRf> The lower limit of the horizontal offset setting range.

## Examples

DISPLAY:FVTIME:MEASview1:X:SCALE:OFFSET:MINIMUM? might return -16.28E-3, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

## DISPlay:FVTime[:MEASview<y>]:Y[:SCALe]

Sets or queries the vertical range of the Frequency versus Time graph.

## Conditions

Measurement views: Frequency versus Time

## Group

Display commands

## Syntax

```
DISPlay:FVTime[:MEASview<y>]:Y[:SCALe] <value>
```

```
DISPlay:FVTime[:MEASview<y>]:Y[:SCALe]?
```

## Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:Y\[:SCALe\]:OFFSet](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the vertical range. Range: 10 Hz to 120 MHz.

## Returns

<NRf>

## Examples

`DISPlay:FVTime:MEASview1:Y:SCALE 30MHz` sets the vertical range to 30 MHz in the Frequency versus Time graph.

## DISPlay:FVTime[:MEASview<y>]:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Frequency versus Time view.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

`DISPlay:FVTime[:MEASview<y>]:Y[:SCALE]:AUTO`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

`DISPLAY:FVTIME:MEASview1:Y:SCALE:AUTO` sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:FVTime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the Frequency versus Time graph.

### Conditions

Measurement views: Frequency versus Time

### Group

Display commands

### Syntax

`DISPlay:FVTime[:MEASview<y>]:Y[:SCALE]:OFFSet <value>`

`DISPlay:FVTime[:MEASview<y>]:Y[:SCALE]:OFFSet?`

### Related Commands

[DISPlay:FVTime\[:MEASview<y>\]:Y\[:SCALE\]](#)

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>: : <NRf> specifies the vertical offset. Range: -60 MHz to +60 MHz.

### Returns

<NRf>

### Examples

DISPLAY:FVTIME:MEASview1:Y:SCALE:OFFSET -14.5MHz sets the vertical offset to -14.5 MHz in the Frequency versus Time graph.

## DISPlay:GENeral:MEASview<y>:DELeTe (No Query Form)

Deletes a measurement view in the general signal viewing.

### Conditions

Measurement views: General signal viewing

### Group

Display commands

### Syntax

DISPlay:GENeral:MEASview<y>:DELeTe { SPECTrum | DPX | AVTime | FVTime | PHVTime | IQVTime | SGRam | TOVerview }

### Arguments

The following table shows the arguments. The arguments are the string type.

**Table 38: General signal viewing views**

Argument	View
SPECTrum	Spectrum
DPX	DPX (Digital Phosphor) spectrum
AVTime	Amplitude versus Time
FVTime	Frequency versus Time
PHVTime	Phase versus Time
IQVTime	IQ versus Time
SGRam	Spectrogram
TOVerview	Time overview

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

`DISPLAY:GENERAL:MEASVIEW1:DELETE` DPX deletes the DPX spectrum view with Measurement View 1.

## DISPlay:GENeral:MEASview<y>[:SOURce<x>]:NEW (No Query Form)

Displays a new measurement view in the general signal viewing.

### Conditions

Measurement views: General signal viewing with specified Measurement view number.

### Group

Display commands

### Syntax

```
DISPlay:GENeral:MEASview<y>[:SOURce<x>]:NEW { SPECTrum | DPX | AVTime | FVTime |
PHVTime | IQVTime | SGRam | TOVerview }
```

### Arguments

(See Table 37 on page 2-566.) If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

<x> = 1 to 8, representing Range of Source.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Examples

`DISPlay:GENeral:MEASview1:SOURce1:NEW DPX` creates the DPX spectrum view for source1.

## DISPlay:GENeral:MEASview<y>:SElect

Selects a measurement view in the general signal viewing on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the `DISPlay:WINDow:OPTimized:MEASurement?` query.

### Conditions

Measurement views: General signal viewing

### Group

Display commands

### Syntax

```
DISPlay:GENeral:MEASview<y>:SElect { SPECTrum | DPX | AVTime | FVTime | PHVTime |
IQVTime | SGRam | TOVerview }
```

```
DISPlay:GENeral:MEASview<y>:SElect?
```

## Arguments

(See Table 37 on page 2-566.) If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

The time overview (TOVerview) cannot be selected as the primary measurement. If you select it, the error (-200, "Execution error; Time Overview cannot be Primary measurement") will be returned. If you use the DISPLAY:GENERAL:MEASview:SElect? query with the time overview as the only measurement active, the error (-200, "Execution error; Analysis selected is not running") will be returned.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

See Arguments.

## Examples

```
DISPLAY:GENERAL:MEASVIEW1:SELECT DPX selects the DPX spectrum view with Measurement view 1.
```

## DISPlay:GPRF:MEASview:DElete (No Query Form)

Deletes a selected measurement view in the RF measurements.

## Conditions

Measurement views: RF measurements

## Group

Display commands

## Syntax

```
DISPlay:GPRF:MEASview:DElete { AM | FM | PM | CCDF | ACPower | FSETtling | MCPower  
| OBW | PNOise | PSETtling | SPURious | SEM }
```

## Arguments

AM deletes the AM view.

FM deletes the AM view.

PM deletes the AM view.

CCDF deletes the CCDF view.

ACPower deletes the Channel power and ACPR view.

FSETtling deletes the Frequency Settling view.

MCPower deletes the MCPR view.

OBW deletes the Occupied Bandwidth view.

PNOise deletes the Phase Noise view (Option 11 only).

PSETtling deletes the Phase Settling view.

SPURious deletes the Spurious view.

SEM deletes the Spectral Emissions Mask view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

DISPlay:GPRF:MEASview:DELeTe ACPower deletes the Channel power and ACPR view.

## DISPlay:GPRF:MEASview:NEW (No Query Form)

Displays a new measurement view in the RF measurements.

## Conditions

Measurement views: RF measurements

## Group

Display commands

## Syntax

```
DISPlay:GPRF:MEASview:NEW { AM | FM | PM | CCDF | ACPower | FSEtTling MCPower | OBW
| PNOise | PSEtTling | SPURious |SEM }
```

## Arguments

AM opens the AM view.

FM opens the AM view.

PM opens the AM view.

CCDF opens the CCDF view.

ACPower opens the Channel power and ACPR view.

FSEtTling opens the Frequency Settling view.

MCPower opens the MCPR view.

OBW opens the Occupied Bandwidth view.

PNOise opens the Phase Noise view (Option 11 only).

PSEtTling opens the Phase Settling view.

SPURious opens the Spurious view.

SEM opens the Spectral Emissions Mask view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

DISPlay:GPRF:MEASview:NEW ACPower creates the Channel power and ACPR view.

## DISPlay:GPRF:MEASview:SElect

Selects a measurement view in the RF measurements on the screen. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

### Conditions

Measurement views: RF measurements

### Group

Display commands

### Syntax

```
DISPlay:GPRF:MEASview:SElect { AM | FM | PM | CCDF | ACPower | FSEtting | MCPower  
| OBW | PNOise | PSEtting | SPURious | SEM }
```

```
DISPlay:GPRF:MEASview:SElect?
```

### Arguments

AM selects the AM view.

FM selects the AM view.

PM selects the AM view.

CCDF selects the CCDF view.

ACPower selects the Channel power and ACPR view.

FSEtting selects the Frequency Settling view.

MCPower selects the MCPR view.

OBW selects the Occupied Bandwidth view.

PNOise selects the Phase Noise view (Option 11 only).

PSEtting selects the Phase Settling view.

SPURious selects the Spurious view.

SEM selects the Spectral Emissions Mask view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

### Returns

See Arguments.

### Examples

```
DISPlay:GPRF:MEASview:SElect ACPower selects the Channel power and ACPR view.
```



## DISPlay:IQVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Display commands

### Syntax

```
DISPlay:IQVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:IQVTime[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATe?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

### Returns

See Arguments.

### Examples

DISPLAY:IQVTIME:MEASview1:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the RF I&Q versus Time view.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]

Sets or queries the horizontal scale (full-scale time) of the RF I&Q versus Time graph. Programming a specified scale sets

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE] AUTO:STATe OFF.
```

### Conditions

Measurement views: RF I&Q versus Time

### Group

Display commands

### Syntax

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE] <value>
```

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]?
```

## Related Commands

[DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe](#), [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <NRF> | MAXimum | MINimum } specifies the horizontal scale in full-scale time. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively. Use the [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:MAXimum?](#) and [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:MINimum?](#) queries to get the upper and lower limit values of the setting range.

## Returns

See Arguments.

## Examples

DISPLAY:IQVTIME:MEASview<y>:X:SCALE 100us sets the horizontal scale to 100  $\mu$ s.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the RF I&Q versus Time view. Executing this command sets `DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe ON`.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO

## Related Commands

[DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]](#), [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:AUTO sets the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE { OFF | ON | 0 | 1 }
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATE?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]](#) and [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#) command.

ON or 1 specifies that the horizontal scale is set automatically.

## Returns

See Arguments.

## Examples

```
DISPLAY:IQVTIME:MEASview1:X:SCALE:AUTO:STATE ON specifies that the horizontal scale is set automatically.
```

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:MAXimum?
```

## Related Commands

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<NRf> The upper limit of the horizontal scale setting range.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:MAXIMUM? might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:MINimum?

## Related Commands

DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<NRf> The lower limit of the horizontal scale setting range.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:MINIMUM? might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the RF I&Q versus Time graph. Programming a specified offset sets  
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE] AUTO:STATe OFF.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet <value>
```

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATE](#), [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <Nrf> | MAXimum | MINimum } specifies the horizontal offset. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet:MAXimum?](#) and [DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet:MINimum?](#) queries to get the upper and lower limit values of the setting range.

## Returns

See Arguments.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:OFFSET 800ns sets the minimum horizontal value to 800 ns in the RF I&Q versus Time graph.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum? (Query Only)

Queries the upper limit of the horizontal offset setting range.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MAXimum?
```

## Related Commands

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

```
DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet
```

## Returns

<Nrf> The upper limit of the horizontal offset setting range.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:OFFSET:MAXIMUM? might return -1.812E-3, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

## DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MINimum? (Query Only)

Queries the lower limit of the horizontal offset setting range.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Display commands

### Syntax

DISPlay:IQVTime[:MEASview<y>]:X[:SCALE]:OFFSet:MINimum?

### Related Commands

[DISPlay:IQVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

### Returns

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<NRf> The lower limit of the horizontal offset setting range.

## Examples

DISPLAY:IQVTIME:MEASview1:X:SCALE:OFFSET:MINIMUM? might return -16.28E-3, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

## DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]

Sets or queries the vertical range of the RF I&Q versus Time graph.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Display commands

### Syntax

DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE] <value>

DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]?

## Related Commands

*DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:OFFSet*

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the vertical range. Range: 1 $\mu$  to 10 V.

## Returns

<NRf>

## Examples

DISPLAY:IQVTIME:MEASview1:Y:SCALE 1.5 sets the vertical range to 1.5 V in the RF I&Q versus Time graph.

## DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the RF I&Q versus Time view.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:AUTO

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

DISPLAY:IQVTIME:MEASview1:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the RF I&Q versus Time graph.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:IQVTime\[:MEASview<y>\]:Y\[:SCALE\]](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value> : :<NRf> specifies the vertical offset. Range: -5 to +5 V.

## Returns

<NRf>

## Examples

DISPLAY:IQVTIME:MEASview1:Y:SCALE:OFFSET -82.75mV sets the vertical offset to -82.75 mV in the RF I&Q versus Time graph.

## DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:RESCale (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Display commands

## Syntax

```
DISPlay:IQVTime[:MEASview<y>]:Y[:SCALE]:RESCale
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

DISPLAY:IQVTIME:MEASview1:Y:SCALE:RESCALE rescales the vertical axis automatically to fit the waveform to the screen.



## DISPlay:LTE:ACLR:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the LTE ACLR display. It will enable or disable the checkbox in the Prefs tab of the LTE ACLR settings control panel. This command will not add the marker itself. It will only decide if the enabled marker readout is shown or hidden.

### Conditions

Measurement view: LTE ACLR

### Group

Display commands

### Syntax

```
DISPlay:LTE:ACLR:MARKer:SHOW:STATe {OFF | ON | 1 | 0}
```

```
DISPlay:LTE:ACLR:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the readout for the selected marker in the LTE ACLR display.

ON or 1 shows the readout for the selected marker in the LTE ACLR display.

### Returns

OFF or 0 means the readout for the selected marker in the LTE ACLR display is hidden.

ON or 1 means the readout for the selected marker in the LTE ACLR display is showing.

### Examples

DISPLAY:LTE:ACLR:MARKER:SHOW:STATE ON shows the readout for the selected marker in the display if the marker is enabled.

## DISPlay:LTE:ACLR:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale of the LTE ACLR display graph to the default values.

### Conditions

Measurement view: LTE ACLR

### Group

Display commands

### Syntax

```
DISPlay:LTE:ACLR:RESet:SCALE
```

### Arguments

None

## Examples

`DISPLAY:LTE:ACLR:RESET:SCALE` resets the horizontal and vertical scale of the graph to the default values.

## DISPlay:LTE:ACLR:SHOW:LIMit

Sets or queries the appearance of the limits in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Display commands

### Syntax

```
DISPlay:LTE:ACLR:SHOW:LIMit { SHADed | LINE | OFF }
```

```
DISPlay:LTE:ACLR:SHOW:LIMit?
```

### Arguments

SHADed: denotes the limit area with shading.

LINE: delineates the limit area with a line.

OFF: turns off limits in the display.

### Returns

SHADED: means that the limit area is denoted with shading.

LINE: means that the limit area is delineated with a line.

OFF: means that the display of limits is off.

## Examples

`DISPLAY:LTE:ACLR:SHOW:LIMIT SHAD` sets the instrument to use shading to identify the limits of the graph in the display.

## DISPlay:LTE:ACLR:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticules in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Display commands

### Syntax

```
DISPlay:LTE:ACLR:WINDow:TRACe:GRATicule:GRID:STATe {OFF | ON | 1 | 0}
```

```
DISPlay:LTE:ACLR:WINDow:TRACe:GRATICule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticules in the graph.

ON or 1 shows the graticules in the graph.

### Returns

0 indicates that the graticules in the graph is hidden.

1 indicates that the graticules in the graph is showing.

### Examples

```
DISPLAY:LTE:ACLR:WINDOW:TRACE:GRATICULE:GRID:STATE OFF hides the graticules in the graph.
```

## DISPlay:LTE:ACLR:WINDow:TRACe:POWER:LEVEL:STATe

Determines whether to show or hide the power readout in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Display commands

### Syntax

```
DISPlay:LTE:ACLR:WINDow:TRACe:POWER:LEVEL:STATe {OFF | ON | 1 | 0}
```

```
DISPlay:LTE:ACLR:WINDow:TRACe:POWER:LEVEL:STATe?
```

### Arguments

OFF or 0 hides the power readout in the display.

ON or 1 shows the power readout in the display.

### Returns

0 indicates that the power readout in the display is hidden.

1 indicates that the power readout in the display is showing.

### Examples

DISPLAY:LTE:ACLR:WINDOW:TRACE:POWER:LEVEL:STATE? might return 1, indicating that the power readout in the display is showing.

## DISPlay:LTE:ACLR:X:SCALE

Sets or queries the horizontal range of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:X:SCALE <value>
```

```
DISPlay:LTE:ACLR:X:SCALE?
```

## Arguments

<value>::<NRf> specifies the horizontal range

## Returns

<NR3>

## Examples

DISPLAY:LTE:ACLR:X:SCALE 10.000000000E+6 sets the horizontal range to 10 MHz.

## DISPlay:LTE:ACLR:X:SCALE:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:X:SCALE:AUTO
```

## Arguments

None

## Examples

DISPLAY:LTE:ACLR:X:SCALE:AUTO rescales the horizontal axis to fit the display screen.

## DISPlay:LTE:ACLR:X:SCALE:OFFSet

Sets or queries the center frequency of the trace in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:X:SCALE:OFFSet <value>
```

```
DISPlay:LTE:ACLR:X:SCALE:OFFSet?
```

## Arguments

<value> ::= <NRf> specifies the center frequency.

## Returns

<NR3>

## Examples

DISPLAY:LTE:ACLR:X:SCALE:OFFSET 1.450000000E+9 sets the center frequency to 1.45 GHz.

## DISPlay:LTE:ACLR:Y:SCALE:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform on the screen in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:Y:SCALE:AUTO
```

## Arguments

None

## Examples

DISPLAY:LTE:ACLR:Y:SCALE:AUTO rescales the vertical scale to fit the waveform on the screen.

## DISPlay:LTE:ACLR:Y:SCALE:OFFSet

Sets or queries the vertical offset of the trace in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:Y:SCALE:OFFSet <value>
```

```
DISPlay:LTE:ACLR:Y:SCALE:OFFSet?
```

## Arguments

<value>::=<NRf> specifies the vertical offset.

## Returns

<NRf>

## Examples

DISPLAY:LTE:ACLR:Y:SCALE:OFFSET 40 sets the vertical position of the trace to 40 dBm.

## DISPlay:LTE:ACLR:Y:SCALE

Sets or queries the vertical range of the graph in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Display commands

## Syntax

```
DISPlay:LTE:ACLR:Y:SCALE <value>
```

```
DISPlay:LTE:ACLR:Y:SCALE?
```

## Arguments

<value>::=<NRf> specifies the vertical range.

## Returns

<NRf>

## Examples

DISPLAY:LTE:ACLR:Y:SCALE 100 sets the vertical range to 100 dB.

## DISPlay:LTE:CHSPectrum:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the LTE Channel Spectrum display. It will enable or disable the checkbox in the Prefs tab of the LTE Channel Spectrum settings control panel. This command will not add the marker itself. It will only decide if the enabled marker readout is shown or hidden.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Display commands

### Syntax

```
DISPlay:LTE:CHSPectrum:MARKer:SHOW:STATe {OFF | ON | 1 | 0}
DISPlay:LTE:CHSPectrum:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the readout for the selected marker in the LTE Channel Spectrum display.

ON or 1 shows the readout for the selected marker in the LTE Channel Spectrum display.

### Returns

OFF or 0 means the readout for the selected marker in the LTE Channel Spectrum display is hidden.

ON or 1 means the readout for the selected marker in the LTE Channel Spectrum display is showing.

### Examples

DISPLAY:LTE:CHSPECTRUM:MARKER:SHOW:STATE ON shows the readout for the selected marker in the display if the marker is enabled.

## DISPlay:LTE:CHSPectrum:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale of the LTE Channel Spectrum display graph to the default values.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Display commands

### Syntax

```
DISPlay:LTE:CHSPectrum:RESet:SCALE
```

### Arguments

None

## Examples

`DISPLAY:LTE:CHSPECTRUM:RESET:SCALE` resets the horizontal and vertical scale of the graph to the default values.

## **DISPlay:LTE:CHSPepectrum:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule in the LTE Channel Spectrum display.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Display commands

### Syntax

`DISPlay:LTE:CHSPepectrum:WINDow:TRACe:GRATicule:GRID:STATe {OFF | ON | 1 | 0}`

`DISPlay:LTE:CHSPepectrum:WINDow:TRACe:GRATicule:GRID:STATe?`

### Arguments

OFF or 0 hides the graticule in the LTE Channel Spectrum display.

ON or 1 shows the graticule in the LTE Channel Spectrum display.

### Returns

OFF or 0 means the graticule in the LTE Channel Spectrum display is hidden.

ON or 1 means the graticule in the LTE Channel Spectrum display is showing.

## Examples

`DISPLAY:LTE:CHSPECTRUM:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` hides the graticules in the LTE Channel Spectrum display.

## **DISPlay:LTE:CHSPepectrum:X:SCALe**

Sets or queries the horizontal range of the LTE Channel Spectrum display.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Display commands

### Syntax

`DISPlay:LTE:CHSPepectrum:X:SCALe <value>`

`DISPlay:LTE:CHSPepectrum:X:SCALe?`



## Arguments

<value> : : <NRf> specifies the horizontal range

## Returns

<NR3>

## Examples

DISPLAY:LTE:CHSPECTRUM:X:SCALE 10.000000000E+6 sets the horizontal range to 10 MHz.

## DISPLay:LTE:CHSPepectrum:X:SCALe:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Display commands

## Syntax

DISPLay:LTE:CHSPepectrum:X:SCALe:AUTO

## Arguments

None

## Examples

DISPLAY:LTE:CHSPECTRUM:X:SCALE:AUTO rescales the horizontal axis to fit the display screen.

## DISPLay:LTE:CHSPepectrum:X:SCALe:OFFSet

Sets or queries the center frequency of the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Display commands

## Syntax

DISPLay:LTE:CHSPepectrum:X:SCALe:OFFSet <value>

DISPLay:LTE:CHSPepectrum:X:SCALe:OFFSet?

## Arguments

<value> ::= <NRf> specifies the center frequency.

## Returns

<NR3>

## Examples

DISPLAY:LTE:ACLR:X:SCAEL:OFFSET 1.45000000E+9 sets the center frequency to 1.45 GHz.

## DISPlay:LTE:CHSPectrum:Y:SCALe

Sets or queries the vertical range of the graph in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Display commands

## Syntax

DISPlay:LTE:CHSPectrum:Y:SCALe <value>

DISPlay:LTE:CHSPectrum:Y:SCALe?

## Arguments

<value> ::= <NRf> specifies the vertical range.

## Returns

<NRf>

## Examples

DISPLAY:LTE:CHSPECTRUM:Y:SCALE 100 sets the vertical range to 100 dB.

## DISPlay:LTE:CHSPectrum:Y:SCALe:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform on the screen in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Display commands

## Syntax

```
DISPlay:LTE:CHSPepectrum:Y:SCALe:AUTO
```

```
DISPlay:LTE:CHSPepectrum:Y:SCALe:AUTO?
```

## Arguments

None

## Examples

DISPLAY:LTE:CHSPECTRUM:Y:SCALE:AUTO rescales the vertical scale to fit the waveform on the screen.

## DISPlay:LTE:CHSPepectrum:Y:SCALe:OFFSet

Sets or queries the vertical offset of the trace in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Display commands

## Syntax

```
DISPlay:LTE:CHSPepectrum:Y:SCALe:OFFSet <value>
```

```
DISPlay:LTE:CHSPepectrum:Y:SCALe:OFFSet?
```

## Arguments

<value>::=<Nrf> specifies the vertical offset.

## Returns

<Nrf>

## Examples

DISPLAY:LTE:CHSPECTRUM:Y:SCALE:OFFSET 40 sets the vertical position of the trace to 40 dBm.

## DISPlay:LTE:CONSte:SCALe

Sets or queries the scale (zoom) in the LTE Constellation display.

## Conditions

Measurement view: LTE Constellation

## Group

Display commands

## Syntax

DISPlay:LTE:CONStE:SCALe <value>

DISPlay:LTE:CONStE:SCALe?

## Arguments

<value>::<Nrf> specifies the scale (zoom)

## Returns

<Nrf>

## Examples

DISPLAY:LTE:CONSTE:SCALE 5 zooms the display out by 5.

## DISPlay:LTE:CONStE:X:OFFSet

Sets or queries the horizontal offset in the LTE Constellation display.

## Conditions

Measurement view: LTE Constellation

## Group

Display commands

## Syntax

DISPlay:LTE:CONStE:X:OFFSet <value>

DISPlay:LTE:CONStE:X:OFFSet?

## Arguments

<value>::<Nrf> specifies the horizontal offset range.

## Returns

<Nrf>

## Examples

DISPLAY:LTE:CONSTE:X:OFFSET 2.7 sets the horizontal offset to 2.7 units to the left.

## DISPlay:LTE:CONStE:Y:OFFSet

Sets or queries the vertical offset in the LTE Constellation display.

## Conditions

Measurement view: LTE Constellation

## Group

Display commands

## Syntax

```
DISPlay:LTE:CONStE:Y:OFFSet <value>
```

```
DISPlay:LTE:CONStE:Y:OFFSet?
```

## Arguments

<value> : <NRf> specifies the vertical offset range. A positive value moves the offset up. A negative value moves the offset down and the display moves in the other direction.

## Returns

<NRf>

## Examples

DISPLAY:LTE:CONSTE:Y:OFFSET 2 sets the vertical offset to 2 units.

## DISPlay:LTE:MEASview:DElete (No Query Form)

Closes the selected LTE measurement display in the application.

## Conditions

Measurement view: Any LTE display

## Group

Display commands

## Syntax

```
DISPlay:LTE:MEASview:DElete { CONS | ACLR | CHSP | PVT }
```

## Arguments

CONS closes the LTE Constellation display.

ACLR closes the LTE ACLR display.

CHSP closes the LTE Channel Spectrum display.

PVT closes the LTE Power vs Time display.

## Examples

DISPLAY:LTE:MEASVIEW:DELETE ACLR closes the LTE ACLR display.

## DISPlay:LTE:MEASview:NEW (No Query Form)

Opens a new LTE measurement display.

## Conditions

Measurement view: Any LTE display

## Group

Display commands

## Syntax

```
DISPlay:LTE:MEASview:NEW { CONS | ACLR | CHSP | PVT }
```

## Arguments

CONS opens the LTE Constellation display.

ACLR opens the LTE ACLR display.

CHSP opens the LTE Channel Spectrum display.

PVT opens the LTE Power vs Time display.

## Examples

```
DISPLAY:LTE:MEASVIEW:NEW ACLR opens a new LTE ACLR display.
```

## DISPlay:LTE:MEASview:SElect

Selects an LTE measurement display in the application. The query returns the currently selected display.

## Conditions

Measurement view: Any LTE display

## Group

Display commands

## Syntax

```
DISPlay:LTE:MEASview:SElect { CONS | ACLR | CHSP | PVT }
```

```
DISPlay:LTE:MEASview:SElect?
```

## Arguments

CONS selects the LTE Constellation display.

ACLR selects the LTE ACLR display.

CHSP selects the LTE Channel Spectrum display. PVT selects the LTE Power vs Time display.

## Returns

CONS means the LTE Constellation display is selected.

ACLR means the LTE ACLR display is selected.

CHSP means the LTE Channel Spectrum display is selected.

PVT means the LTE Power vs Time display is selected.

## Examples

`DISPLAY:LTE:MEASVIEW:SELECT ACLR` selects the LTE ACLR display.

## DISPlay:LTE:PVTime:MARKer:SHOW:STATe

Shows or hides the marker readout on the LTE Power vs. Time display.

### Conditions

Measurement view: LTE Power vs Time

### Group

Display commands

### Syntax

```
DISPlay:LTE:PVTime:MARKer:SHOW:STATe {OFF | ON | 0 | 1 }
```

```
DISPlay:LTE:PVTime:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the marker readout in the LTE Power vs Time display.

ON or 1 shows the marker readout in the LTE Power vs Time display.

### Returns

OFF or 0 means the marker readout in the LTE Power vs Time display is hidden.

ON or 1 means the marker readout in the LTE Power vs Time display is showing.

## Examples

`DISPLAY:LTE:PVTIME:MARKER:SHOW:STATE ON` turns on the marker readout.

## DISPlay:LTE:PVTime:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the LTE Power vs Time display.

### Conditions

Measurement view: LTE Power vs Time

### Group

Display commands

### Syntax

```
DISPlay:LTE:PVTime:WINDow:TRACe:GRATicule:GRID:STATe {OFF | ON | 0 | 1 }
```

```
DISPlay:LTE:PVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the LTE Power vs Time display.

ON or 1 shows the graticule in the LTE Power vs Time display.

## Returns

OFF or 0 means the graticule in the LTE Power vs Time display is hidden.

ON or 1 means the graticule in the LTE Power vs Time display is showing.

## Examples

`DISPLAY:LTE:PVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` hides the graticules in the LTE Power vs Time display.

## DISPlay:LTE:PVTime:X:SCALE

Sets or queries the vertical range of the LTE Power vs Time graph.

## Conditions

Measurement view: LTE Power vs Time

## Group

Display commands

## Syntax

`DISPlay:LTE:PVTime:X:SCALE <NRf>`

`DISPlay:LTE:PVTime:X:SCALE?`

## Arguments

<NRf> specifies the vertical range.

## Returns

<NRf>

## Examples

`DISPLAY:LTE:PVTIME:X:SCALE 10` sets the horizontal range to 10 dB.

## DISPLay:LTE:PVTime:X:SCALE:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time



## Group

Display commands

## Syntax

```
DISPLay:LTE:PVTime:X:SCALE:AUTO
```

## Arguments

None

## Examples

DISPLAY:LTE:PVTIME:X:SCALE:AUTO rescales the horizontal axis to fit the display screen.

## DISPlay:LTE:PVTime:X:SCALE:RESet (No Query Form)

Resets the horizontal scale to the default value for the time measurements for the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

## Group

Display commands

## Syntax

```
DISPlay:LTE:PVTime:X:SCALE:RESet
```

## Arguments

None

## Examples

DISPLAY:LTE:PVTIME:X:SCALE:RESET resets the horizontal scale to the default value.

## DISPlay:LTE:PVTime:Y:SCALE

Sets or queries the horizontal range of the LTE Power vs Time graph.

## Conditions

Measurement view: LTE Power vs Time

## Group

Display commands

## Syntax

```
DISPlay:LTE:PVTime:Y:SCALE <value>
```

DISPlay:LTE:PVTime:Y:SCALE?

### Arguments

<value> : : <NRf> specifies the horizontal range.

### Returns

<NRf>

### Examples

DISPLAY:LTE:PVTIME:Y:SCALE 100 sets the horizontal range to 100 ms.

## DISPlay:LTE:PVTime:Y:SCALE:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform on the screen in the LTE Power vs Time display.

### Conditions

Measurement view: LTE Power vs Time

### Group

Display commands

### Syntax

DISPlay:LTE:PVTime:Y:SCALE:AUTO

### Arguments

None

### Examples

DISPLAY:LTE:PVTIME:Y:SCALE:AUTO rescales the vertical scale to fit the waveform on the screen.

## DISPlay:LTE:PVTime:Y:SCALE:OFFSet

Sets or queries the vertical offset (top edge of the vertical axis) of the trace in the LTE Power vs Time display.

### Conditions

Measurement view: LTE Power vs Time

### Group

Display commands

### Syntax

DISPlay:LTE:PVTime:Y:SCALE:OFFSet <NRf>

DISPlay:LTE:PVTime:Y:SCALE:OFFSet?

## Arguments

<NRf> specifies the vertical offset.

## Returns

<NRf>

## Examples

DISPLAY:LTE:PVTIME:Y:SCALE:OFFSET -10 sets the vertical position of the trace to -10 dBm.

## DISPlay:LTE:PVTime:Y:SCALE:PDIVision

Sets or queries the value of the vertical scale for the LTE Power vs. Time display in dB/division. This is only a visual control for panning the graph.

## Conditions

Measurement view: LTE Power vs Time

## Group

Display commands

## Syntax

DISPlay:LTE:PVTime:Y:SCALE:PDIVision <NRf>

DISPlay:LTE:PVTime:Y:SCALE:PDIVision?

## Arguments

<NRf> specifies the vertical scale in dB/div.

## Returns

<NRf>

## Examples

DISPLAY:LTE:PVTIME:Y:SCALE:PDIVISION 20 sets the vertical scale to 20.0 dB/div.

## DISPlay:LTE:PVTime:Y:SCALE:RESet (No Query Form)

Resets the vertical scale and position (offset) to the default value for the time measurements for the LTE Power vs Time display.

## Conditions

Measurement view: LTE Power vs Time

## Group

Display commands

## Syntax

DISPlay:LTE:PVTime:Y:SCALE:RESet

## Arguments

None

## Examples

DISPLAY:LTE:PVTIME:Y:SCALE:RESET resets the vertical scale and position (offset) to the default value.

## DISPlay:MCPower:MARKer:SHOW:STATE

Determines whether to show or hide the readout for the selected marker in the MCPR view.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

DISPlay:MCPower:MARKer:SHOW:STATE { OFF | ON | 0 | 1 }

DISPlay:MCPower:MARKer:SHOW:STATE?

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

DISPLAY:MCPOWER:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.

## DISPlay:MCPower:PLEVel:SHOW:STATE

Determines whether to show or hide the power levels in the MCPR view.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

DISPlay:MCPower:PLEVel:SHOW:STATE { OFF | ON | 0 | 1 }

DISPlay:MCPower:PLEVel:SHOW:STATE?

## Arguments

OFF or 0 hides the power levels in the graph.

ON or 1 shows the power levels in the graph.

## Examples

`DISPLAY:MCPOWER:PLEVEL:SHOW:STATE ON` shows the power levels in the graph.

## DISPlay:MCPower:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the MCPR view.

Vertical offset = Reference level, Vertical scale = 100 dB,

Horizontal offset = Center frequency, and Horizontal scale = Default span

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

`DISPlay:MCPower:RESet:SCALE`

## Arguments

None

## Examples

`DISPLAY:MCPOWER:RESET:SCALE` resets the horizontal and vertical scale to the default values.

## DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

`DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }`

`DISPlay:MCPower:WINDow:TRACe:GRATicule:GRID:STATe?`

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

`DISPLAY:MCPOWER:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

## DISPlay:MCPower:X[:SCALE]

Sets or queries the horizontal range of the MCPR graph.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

`DISPlay:MCPower:X[:SCALE] <value>`

`DISPlay:MCPower:X[:SCALE]?`

## Related Commands

[\*DISPlay:MCPower:X\[:SCALE\]:OFFSet\*](#)

## Arguments

`<value>::<NRf>` specifies the horizontal range.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`DISPLAY:MCPOWER:X:SCALE 10MHz` sets the horizontal range to 10 MHz.

## DISPlay:MCPower:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the MCPR view.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

```
DISPlay:MCPower:X[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:MCPOWER:X:SCALE:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:MCPower:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the MCPR graph.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

```
DISPlay:MCPower:X[:SCALE]:OFFSet <value>
```

```
DISPlay:MCPower:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:MCPower:X\[:SCALE\]](#)

## Arguments

<value> : : <Nrf> specifies the minimum horizontal value.

Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

## Examples

DISPLAY:MCPOWER:X:SCALE:OFFSET 1.45GHz sets the minimum horizontal value to 1.45 GHz in the MCPR graph.

## DISPlay:MCPower:Y[:SCALE]

Sets or queries the vertical range of the MCPR graph.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

```
DISPlay:MCPower:Y[:SCALE] <value>
```

```
DISPlay:MCPower:Y[:SCALE]?
```

## Related Commands

[DISPlay:MCPower:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::<NRf> specifies the vertical range. Range: 0.1 to 200 dB.

## Examples

DISPLAY:MCPOWER:Y:SCALE 100 sets the vertical range to 100 dB in the MCPR graph.

## DISPlay:MCPower:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the MCPR view.

## Conditions

Measurement views: MCPR

## Group

Display commands

## Syntax

```
DISPlay:MCPower:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:MCPOWER:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:MCPower:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the MCPR graph.

## Conditions

Measurement views: MCPR



## Group

Display commands

## Syntax

```
DISPlay:MCPower:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:MCPower:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:MCPower:Y\[:SCALE\]](#)

## Arguments

<value>::<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

## Examples

DISPLAY:MCPOWER:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBm in the MCPR graph.

## DISPlay:MERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule grid view state for the Magnitude error versus Time view.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Display commands

## Syntax

```
DISPlay:MERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:MERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

OFF or 0

ON or 1

## Examples

`DISPlay:MERRor:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the Magnitude error versus Time Measview 2.

## **DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]**

Sets or queries the vertical range of the Magnitude error versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Display commands

## Syntax

`DISPlay:MERRor[:MEASview<y>]:Y[:SCALe] <value>`

`DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]?`

## Related Commands

[\*DISPlay:MERRor:\[MEASview<y>\]:Y\[:SCALe\]:OFFSet\*](#)

## Arguments

<value>::<NRF> specifies the vertical range.

Range: 1 to 100%.

## Returns

<NRF>

## Examples

`DISPlay:MERRor:MEASview1:Y:SCALe 50` sets the vertical range to 50% in the Magnitude error versus Time display Measurement view 1.

## **DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]:AUTO (No query form)**

Sets the vertical scale automatically to fit the waveform to the screen in the Magnitude error versus Time display.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Display commands

## Syntax

```
DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]:AUTO
```

## Arguments

None

## Examples

DISPLay:MERRor:MEASview2:Y:SCALe:AUTO sets the vertical scale automatically to fit the waveform to the screen of Magnitude error versus Time Measurement view 2.

## DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the Magnitude error versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Display commands

## Syntax

```
DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]:OFFSet <value>
```

```
DISPlay:MERRor[:MEASview<y>]:Y[:SCALe]:OFFSet?
```

## Related Commands

[DISPlay:MERRor\[:MEASview<y>\]:Y\[:SCALe\]](#)

## Arguments

<value>::<NRf> specifies the minimum vertical value.

Range: -100 to +100%.

## Returns

<NRF>

## Examples

`DISPlay:MEERRor:MEASview2:Y:SCALE:OFFSet?-9.5` sets the minimum vertical value to -9.5% in the Magnitude error versus Time graph Measurement View 2.

## DISPlay:NOISe:FIGure[:SCALE]:AUTO (No Query Form)

Automatically scales the Noise Figure graph.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

`DISPlay:NOISe:FIGure[:SCALE]:AUTO`

## Arguments

None

## Examples

`DISP:NOIS:FIG:SCAL:AUTO` will automatically scale the Noise Figure graph.

## DISPlay:NOISe:FIGure:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule state in the Noise Figure display to on (showing) or off (hidden).

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

`DISPlay:NOISe:FIGure:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }`

`DISPlay:NOISe:FIGure:WINDow:TRACe:GRATicule:GRID:STATe?`

## Arguments

OFF or 0 hides the graticule in the Noise Figure display.

ON or 1 shows the graticule in the Noise Figure display.

## Returns

0 means the graticule in the Noise Figure display is hidden (off).

1 means the graticule in the Noise Figure display is showing (on).

## Examples

```
DISPLAY:NOISE:FIGURE:WINDOW:TRACE:GRATICULE:GRID:STATE ON
```

will turn on the graticule in the Noise Figure display.

DISPLAY:NOISE:FIGURE:WINDOW:TRACE:GRATICULE:GRID:STATE? 1 indicates that the graticule in the Noise Figure display is showing (on).

## DISPlay:NOISe:FIGure:WINDow:TRACe:LEGend:STATe

Sets or queries the trace legend state in the Noise Figure display to on (showing) or off (hidden).

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:FIGure:WINDow:TRACe:LEGend:STATe?
```

## Arguments

OFF or 0 hides the trace legend in the Noise Figure display.

ON or 1 shows the trace legend in the Noise Figure display.

## Returns

0 means the trace legend in the Noise Figure display is hidden (off).

1 means the trace legend in the Noise Figure display is showing (on).

## Examples

```
DISPLAY:NOISE:FIGURE:WINDOW:TRACE:LEGEND:STATE ON
```

will turn on the trace legend in the Noise Figure display.

DISPLAY:NOISE:FIGURE:WINDOW:TRACE:LEGEND:STATE? 1 indicates that the trace legend in the Noise Figure display is showing (on).

## DISPlay:NOISe:FIGure:WINDow:TRACe:MEASpoints:STATe

Sets or queries the measurement points state in the Noise Figure display to on (showing) or off (hidden).

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:WINDow:TRACe:MEASpoints:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:FIGure:WINDow:TRACe:MEASpoints:STATe?
```

## Arguments

OFF or 0 hides the measurement points in the Noise Figure display.

ON or 1 shows the measurement points in the Noise Figure display.

## Returns

0 means the measurement points in the Noise Figure display is hidden (off).

1 means the measurement points in the Noise Figure display is showing (on).

## Examples

```
DISPLAY:NOISE:FIGURE:WINDOW:TRACE:MEASPOINTS:STATE ON
```

will turn on the measurement points in the Noise Figure display.

```
DISPLAY:NOISE:FIGURE:WINDOW:TRACE:MEASPOINTS:STATE?
```

might return 1, indicating that the measurement points in the Noise Figure display are showing (on).

## DISPlay:NOISe:FIGure:X:AUTO (No Query Form)

Automatically sets the horizontal scale of the Noise Figure graph.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:X:AUTO
```

## Arguments

None

## DISPlay:NOISe:FIGure:X:OFFSet

Sets or queries the center frequency (offset) in the Noise Figure display.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:X:OFFSet <NRf>
```

```
DISPlay:NOISe:FIGure:X:OFFSet?
```

## Arguments

<NRf> is the center frequency (offset) in the Noise Figure display.

Range: -1.9900 GHz to +5.0000 GHz

## Returns

<NR3> is a floating point value with an exponent. This is the center frequency (offset) value.

## Examples

DISPLAY:NOISE:FIGURE:X:OFFSET? might return 2.0000000009E+9, indicating that the center frequency (offset) value is 2.0000 GHz.

## DISPlay:NOISe:FIGure:X[:SCALe]

Sets or queries the horizontal scale of the Noise Figure display.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:X[:SCALe] <NRf>
```

```
DISPlay:NOISe:FIGure:X[:SCALe]
```

## Arguments

<NRf> is the horizontal scale value of the display.

Range: 100 Hz to 6.2 GHz

## Returns

<NR3> = a floating point value with an exponent. This is the horizontal scale value.

## Examples

`DISPLAY:NOISE:FIGURE:X:SCALE?` might return `2.9900000000E+9`, indicating that the horizontal scale value is 3.0 GHz.

## DISPlay:NOISe:FIGure:Y:AUTO (No Query Form)

Automatically sets the vertical scale of the Noise Figure graph.

### Conditions

Measurement view: Noise Figure

### Group

Display commands

### Syntax

```
DISPlay:NOISe:FIGure:Y:AUTO
```

### Arguments

None

## DISPlay:NOISe:FIGure:Y:AUTO:STATe

Sets the state of the automatic vertical scale and position of the graph to on or off. The query form of the command returns the current state.

### Conditions

Measurement view: Noise Figure

### Group

Display commands

### Syntax

```
DISPlay:NOISe:FIGure:Y:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:FIGure:Y:AUTO:STATe?
```

### Arguments

OFF or 0 sets the analyzer to manual setting of the vertical scale and position of the graph.

ON or 1 sets the analyzer to automatically set the vertical scale and position of the graph.

### Returns

OFF or 0 means the analyzer is in manual mode for setting of the vertical scale and position of the graph.

ON or 1 means the analyzer is in automatic mode for setting of the vertical scale and position of the graph.



## Examples

`DISPLAY:NOISE:FIGURE:Y:AUTO:STATE ON` sets the analyzer to automatically set the vertical scale and position of the graph.

## DISPlay:NOISe:FIGure:Y:PDIVision

Sets or queries the distance between graticule lines on the display in dB.

### Conditions

Measurement view: Noise Figure

### Group

Display commands

### Syntax

```
DISPlay:NOISe:FIGure:Y:PDIVision <NRf>
```

```
DISPlay:NOISe:FIGure:Y:PDIVision?
```

### Arguments

<NRf> is the distance between graticule lines on the display.

Range: 100 mdB to 5.00 dB

### Returns

<NR2> = a floating point value without an exponent, which is the distance between graticule lines on the display in dB.

## Examples

`DISPLAY:NOISE:FIGURE:Y:PDIVISION?` might return 1.5000000000,

indicating that the distance between graticule lines on the display is 1.50 dB.

## DISPlay:NOISe:FIGure:Y:POSition

Sets or queries the vertical position of the Noise Figure display.

### Conditions

Measurement view: Noise Figure

### Group

Display commands

### Syntax

```
DISPlay:NOISe:FIGure:Y:POSition <NRf>
```

```
DISPlay:NOISe:FIGure:Y:POSition?
```

## Arguments

<NRf> is the vertical position of the Noise Figure display.

Range: 5.12 to 26.5

## Returns

<NR2> = a floating point value without an exponent, which is the vertical position value.

The vertical position value

## Examples

DISPLAY:NOISE:FIGURE:Y:POSITION? might return 26.5000000000,  
indicating that the vertical position is 26.5.

## DISPlay:NOISe:FIGure:Y:SCALe

Sets or queries the vertical scale of the Noise Figure display.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

DISPlay:NOISe:FIGure:Y:SCALe <NRf>

DISPlay:NOISe:FIGure:Y:SCALe?

## Arguments

<NRf> is the vertical scale of the Noise Figure display Range: 10.2 to 31.6

## Returns

<NR2> = a floating point value without an exponent, which is the vertical scale value.

## Examples

DISPLAY:NOISE:FIGURE:Y:SCALE? might return 10.0000000000, indicating that the vertical scale is 10.0 dB.

## DISPlay:NOISe:FIGure:Y:SCALe:LINear:STATe

Sets or queries whether the linear units are shown (on) or not shown (off) on the display.

## Conditions

Measurement view: Noise Figure

## Group

Display commands

## Syntax

```
DISPlay:NOISe:FIGure:Y:SCALe:LINear:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:FIGure:Y:SCALe:LINear:STATe?
```

## Arguments

OFF or 0 sets the analyzer to show the linear units on the display.

ON or 1 sets the analyzer to not show the linear units on the display.

## Returns

OFF or 0 indicates that the analyzer is set to show the linear units on the display.

ON or 1 indicates that the analyzer is set to not show the linear units on the display.

## Examples

DISPlay:NOISe:FIGure:Y:SCALe:LINear:STATe? might return 1, indicating that linear units are not showing on the display.

## DISPlay:NOISe:GAIN[:SCALe]:AUTO (No Query Form)

Automatically scales the Gain graph.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

```
DISPlay:NOISe:GAIN[:SCALe]:AUTO
```

## Arguments

None

## DISPlay:NOISe:GAIN:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule state in the Gain display to on (showing) or off (hidden).

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

```
DISPlay:NOISe:GAIN:WINDow:TRACe:GRATicule:GRID:STATe  
{OFF|ON|0|1}  
DISPlay:NOISe:GAIN:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the Gain display.

ON or 1 shows the graticule in the Gain display.

## Returns

0 means the graticule in the Gain display is hidden (off).

1 means the graticule in the Gain display is showing (on).

## Examples

DISPLAY:NOISE:GAIN:WINDOW:TRACE:GRATICULE:GRID:STATE ON will turn on the graticule in the Gain display.

DISPLAY:NOISE:GAIN:WINDOW:TRACE:GRATICULE:GRID:STATE? might return 1, indicating that the graticule in the Gain display is showing (on).

## DISPlay:NOISe:GAIN:WINDow:TRACe:LEGend:STATe

Sets or queries the trace legend state in the Gain display to on (showing) or off (hidden).

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

```
DISPlay:NOISe:GAIN:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }  
DISPlay:NOISe:GAIN:WINDow:TRACe:LEGend:STATe?
```

## Arguments

OFF or 0 hides the trace legend in the Gain display.

ON or 1 shows the trace legend in the Gain display.

## Returns

0 means the trace legend in the Gain display is hidden (off).

1 means the trace legend in the Gain display is showing (on).

## Examples

`DISPLAY:NOISE:GAIN:WINDOW:TRACE:LEGEND:STATE ON` will turn on the trace legend in the Gain display.

`DISPLAY:NOISE:GAIN:WINDOW:TRACE:LEGEND:STATE? 1` indicates that the trace legend in the Gain display is showing (on).

## DISPlay:NOISe:GAIN:WINDow:TRACe:MEASPoints:STATe

Sets or queries the measurement points state in the Gain display to on (showing) or off (hidden).

### Conditions

Measurement view: Gain

### Group

Display commands

### Syntax

`DISPlay:NOISe:GAIN:WINDow:TRACe:MEASPoints:STATe {OFF|ON|0|1}`

`DISPlay:NOISe:GAIN:WINDow:TRACe:MEASPoints:STATe?`

### Arguments

OFF or 0 hides the measurement points in the Gain display.

ON or 1 shows the measurement points in the Gain display.

### Returns

0 means the measurement points in the Gain display is hidden (off).

1 means the measurement points in the Gain display is showing (on).

## Examples

`DISPLAY:NOISE:GAIN:WINDOW:TRACE:MEASPOINTS:STATE ON` will turn on the measurement points in the Gain display.

`DISPLAY:NOISE:GAIN:WINDOW:TRACE:MEASPOINTS:STATE?` might return 1, indicating that the measurement points in the Gain display are showing (on).

## DISPlay:NOISe:GAIN:X:AUTO (No Query Form)

Automatically sets the horizontal scale of the Gain graph.

### Conditions

Measurement view: Gain

### Group

Display commands

## Syntax

DISPlay:NOISe:GAIN:X:AUTO

## Arguments

None

## DISPlay:NOISe:GAIN:X:OFFSet

Sets or queries the center frequency (offset) in the Gain display.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

DISPlay:NOISe:GAIN:X:OFFSet <NRf>

DISPlay:NOISe:GAIN:X:OFFSet?

## Arguments

<NRf> is the center frequency (offset) in the Gain display.

Range: -1.9900 GHz to +5.0000 GHz

## Returns

<NR3> = a floating point value with an exponent, which is the center frequency (offset) value.

## Examples

DISPLAY:NOISE:GAIN:X:OFFSET? might return 2.000000009E+9,

indicating that the center frequency (offset) value is 2.0000 GHz.

## DISPlay:NOISe:GAIN:X:SCALE

Sets or queries the horizontal scale of the Gain display.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

DISPlay:NOISe:GAIN:X:SCALE <NRf>

---

DISPlay:NOISe:GAIN:X:SCALe

## Related Commands

[DISPlay:NOISe:GAIN:Y:SCALe](#)

## Arguments

<NRf> is the horizontal scale of the Gain display.

Range: 100 Hz to 6.2 GHz

## Returns

<NR3> = a floating point value with an exponent, which is the horizontal scale value.

## Examples

DISP:NOIS:GAIN:X:SCAL? might return 2.9900000000E+9, indicating that the horizontal scale value is 3.0 GHz.

## DISPlay:NOISe:GAIN:Y:AUTO (No Query Form)

Automatically sets the vertical scale of the Gain graph.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

DISPlay:NOISe:GAIN:Y:AUTO

## Related Commands

[DISPlay:NOISe:GAIN:Y:AUTO:STATe](#)

[DISPlay:NOISe:GAIN:X:AUTO](#)

## Arguments

None

## DISPlay:NOISe:GAIN:Y:AUTO:STATe

Sets or queries the state of the automatic vertical scale and position of the Gain graph to on or off.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

```
DISPlay:NOISe:GAIN:Y:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:GAIN:Y:AUTO:STATe?
```

## Related Commands

[DISPlay:NOISe:GAIN:Y:AUTO](#)

## Arguments

OFF or 0 sets the analyzer to manual setting of the vertical scale and position of the graph.

ON or 1 sets the analyzer to automatically set the vertical scale and position of the graph.

## Returns

OFF or 0 means the analyzer is in manual mode for setting of the vertical scale and position of the graph.

ON or 1 means the analyzer is in automatic mode for setting of the vertical scale and position of the graph.

## Examples

DISPLAY:NOISE:GAIN:Y:AUTO:STATE ON sets the analyzer to automatically set the vertical scale and position of the graph.

## DISPlay:NOISe:GAIN:Y:PDIVision

Sets or queries the distance between graticule lines on the Gain display in dB.

## Conditions

Measurement view: Gain

## Group

Display commands

## Syntax

```
DISPlay:NOISe:GAIN:Y:PDIVision <NRf>
```

```
DISPlay:NOISe:GAIN:Y:PDIVision?
```

## Related Commands

## Arguments

<NRf> is the distance between graticule lines on the Gain display.

## Returns

<NR2> = a floating point value without exponents, which is the distance between graticule lines on the display in dB.



## Examples

`DISPLAY:NOISE:GAIN:Y:PDIVISION?` might return 1.500000000, indicating that the distance between graticule lines on the display is 1.50 dB.

## DISPlay:NOISe:GAIN:Y:POSition

Sets or queries the vertical position of the Gain display.

### Conditions

Measurement view: Gain

### Group

Display commands

### Syntax

```
DISPlay:NOISe:GAIN:Y:POSition <NRf>
```

```
DISPlay:NOISe:GAIN:Y:POSition?
```

### Related Commands

### Arguments

<NRf> is the vertical position of the Gain display.

Range: 5.12 to 26.5

### Returns

<NR2> = a floating point value without exponents, which is the vertical position value.

## Examples

`DISPLAY:NOISE:GAIN:Y:POSITION?` might return 26.500000000, indicating that the vertical position is 26.5.

## DISPlay:NOISe:GAIN:Y:SCALe

Sets or queries the vertical scale of the Gain display.

### Conditions

Measurement view: Gain

### Group

Display commands

### Syntax

```
DISPlay:NOISe:GAIN:Y:SCALe <NRf>
```

```
DISPlay:NOISe:GAIN:Y:SCALe?
```

## Related Commands

### Arguments

<NRf> is the vertical scale of the Gain display.

Range: 10.2 to 31.6

### Returns

<NR2> = a floating point value without an exponent, which is the vertical scale value.

### Examples

DISPLAY:NOISE:GAIN:Y:SCALE? might return 10.000000000, indicating that the vertical scale is 10.0 dB

## DISPlay:NOISe:MEASview:DELeTe (No Query Form)

Deletes the selected Noise and Gain measurement view.

### Conditions

Measurement views: All Noise and Gain displays

### Group

Display commands

### Syntax

```
DISPlay:NOISe:MEASview:DELeTe { GAIN | YFACTOR | FIGure | TEMPerature | TABLE |  
UNCertainty }
```

## Related Commands

[DISPlay:NOISe:MEASview:SElect](#)

[DISPlay:NOISe:MEASview:NEW](#)

### Arguments

GAIN: Deletes the Gain display from the view.

YFACTOR: Deletes the Y Factor display from the view.

FIGure: Deletes the Noise Figure display from the view.

TEMPerature: Deletes the Noise Temperature display from the view.

TABLE: Deletes the Noise Table display from the view.

UNCertainty: Deletes the Uncertainty Calculator display from the view.

### Examples

DISPLAY:NOISE:MEASVIEW:DELETE TABL will delete the Noise Table display view from the screen.

## DISPlay:NOISe:MEASview:NEW (No Query Form)

Displays a new Noise and Gain measurement view.

### Conditions

Measurement views: All Noise and Gain displays

### Group

Display commands

### Syntax

```
DISPlay:NOISe:MEASview:NEW { GAIN | YFACTOR | FIGure | TEMPerature | TABLE |
UNCertainty }
```

### Related Commands

[DISPlay:NOISe:MEASview:SElect](#)

[DISPlay:NOISe:MEASview:DELeTe](#)

### Arguments

GAIN: Displays the Gain display.

YFACTOR: Displays the Y Factor display.

FIGure: Displays the Noise Figure display.

TEMPerature: Displays the Noise Temperature display.

TABLE: Displays the Noise Table display.

UNCertainty: Displays the Uncertainty Calculator display.

### Examples

DISPLAY:NOISE:MEASVIEW:NEW YFAC will bring the Y Factor display up on the screen.

## DISPlay:NOISe:MEASview:SElect

Selects a noise measurement view. The query command returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement query.

### Conditions

Measurement views: All Noise and Gain displays The measurement view must be displayed.

### Group

Display commands

## Syntax

```
DISPlay:NOISe:MEASview:SElect { GAIN | YFACTOR | FIGure | TEMPerature | TABLE |  
UNCertainty }
```

```
DISPlay:NOISe:MEASview:SElect?
```

## Related Commands

[DISPlay:NOISe:MEASview:DELeTe](#)

[DISPlay:NOISe:MEASview:NEW](#)

## Arguments

**GAIN:** Selects the Gain display.

**YFACTOR:** Selects the Y Factor display.

**FIGure:** Selects the Noise Figure display.

**TEMPerature:** Selects the Noise Temperature display.

**TABLE:** Selects the Noise Table display.

**UNCertainty:** Selects the Uncertainty Calculator display.

## Returns

**GAIN:** Means that the Gain display is selected.

**YFAC:** Means that the Y Factor display is selected.

**FIG:** Means that the Noise Figure display is selected.

**TEMP:** Means that the Noise Temperature display is selected.

**TABL:** Means that the Noise Table display is selected.

**UNC:** Means that the Uncertainty Calculator display is selected.

## Examples

`DISPLAY:NOISE:MEASVIEW:SELECT GAIN` will select the Gain display.

## DISPlay:NOISe:POWer:LINear:STATe

Sets or queries whether the linear units are shown (on) or not shown (off) on the Noise Table display.

## Conditions

Measurement view: Noise Table

## Group

Display commands

## Syntax

```
DISPlay:NOISe:POWer:LINear:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:POWer:LINear:STATe?
```

### Arguments

OFF or 0 sets the analyzer to show the linear units on the display.

ON or 1 sets the analyzer to not show the linear units on the display.

### Returns

OFF or 0 indicates that the analyzer is set to show the linear units on the display.

ON or 1 indicates that the analyzer is set to not show the linear units on the display.

### Examples

DISPLAY:NOISE:POWER:LINEAR:STATE? might return 1, indicating that linear units are not showing on the Noise Table display.

## DISPlay:NOISe:TEMPerature[:SCALe]:AUTO (No Query Form)

Automatically scales the Noise Temperature graph.

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature[:SCALe]:AUTO
```

### Arguments

None

## DISPlay:NOISe:TEMPerature:WINDow:TRACe:GRATICule:GRID:STATe

Sets or queries the graticule state in the Noise Temperature display to on (showing) or off (hidden).

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:GRATICule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:GRATICule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the Noise Temperature display.

ON or 1 shows the graticule in the Noise Temperature display.

## Returns

0 means the graticule in the Noise Temperature display is hidden (off).

1 means the graticule in the Noise Temperature display is showing (on).

## Examples

`DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:GRATICULE:GRID:STATE ON` will turn on the graticule in the Noise Temperature display.

`DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:GRATICULE:GRID:STATE?` might return 1, indicating that the graticule in the Noise Temperature display is showing (on).

## DISPlay:NOISe:TEMPerature:WINDow:TRACe:LEGend:STATe

Sets or queries the trace legend state in the Noise Temperature display to on (showing) or off (hidden).

## Conditions

Measurement view: Noise Temperature

## Group

Display commands

## Syntax

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:LEGend:STATe?
```

## Arguments

OFF or 0 hides the trace legend in the Noise Temperature display.

ON or 1 shows the trace legend in the Noise Temperature display.

## Returns

0 means the trace legend in the Noise Temperature display is hidden (off).

1 means the trace legend in the Noise Temperature display is showing (on).

## Examples

`DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:LEGEND:STATE ON` will turn on the trace legend in the Noise Temperature display.

`DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:LEGEND:STATE?` 1 indicates that the trace legend in the Noise Temperature display is showing (on).

## DISPlay:NOISe:TEMPerature:WINDow:TRACe:MEASPoints:STATe

Sets or queries the measurement points state in the Noise Temperature display to on (showing) or off (hidden).

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:MEASPoints:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:TEMPerature:WINDow:TRACe:MEASPoints:STATe?
```

### Arguments

OFF or 0 hides the measurement points in the Noise Temperature display.

ON or 1 shows the measurement points in the Noise Temperature display.

### Returns

0 means the measurement points in the Noise Temperature display is hidden (off).

1 means the measurement points in the Noise Temperature display is showing (on).

### Examples

DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:MEASPOINTS:STATE ON will turn on the measurement points in the Noise Temperature display.

DISPLAY:NOISE:TEMPERATURE:WINDOW:TRACE:MEASPOINTS:STATE? might return 1, indicating that the measurement points in the Noise Temperature display are showing (on).

## DISPlay:NOISe:TEMPerature:X:AUTO (No Query Form)

Automatically sets the horizontal scale of the Noise Temperature graph.

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature:X:AUTO
```

### Arguments

None

## DISPlay:NOISe:TEMPerature:X:OFFSet

Sets or queries the center frequency (offset) value for the Noise Temperature display.

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature:X:OFFSet <NRf>
```

```
DISPlay:NOISe:TEMPerature:X:OFFSet?
```

### Arguments

<NRf> is the center frequency (offset) value.

Range: -1.9900 GHz to +5.0000 GHz

### Returns

<NR3> = a floating point value with an exponent, which is the center frequency (offset) value.

### Examples

DISPLAY:NOISE:TEMPERATURE:X:OFFSET? might return 2.000000009E+9, indicating that the center frequency (offset) value is 2.0000 GHz.

## DISPlay:NOISe:TEMPerature:X[:SCALE]

Sets or queries the horizontal scale of the Noise Temperature display.

### Conditions

Measurement view: Noise Temperature

### Group

Display commands

### Syntax

```
DISPlay:NOISe:TEMPerature:X[:SCALE] <NRf>
```

```
DISPlay:NOISe:TEMPerature:X[:SCALE]
```

### Related Commands

[DISPlay:NOISe:GAIN:Y:SCALE](#)



## Arguments

<NRf> is the horizontal scale of the Noise Temperature display.

Range: 100 Hz to 6.2 GHz

## Returns

<NR3> = a floating point value with an exponent, which is the horizontal scale value.

## Examples

DISP:NOIS:TEMP:X:SCAL? might return 2.990000000E+9, indicating that the horizontal scale value is 3.0 GHz.

## DISPlay:NOISe:TEMPerature:X:START

Sets or queries the Start Frequency value in the Noise Temperature display.

## Conditions

Measurement views: Noise Temperature

## Group

Display commands

## Syntax

DISPlay:NOISe:TEMPerature:X:START <NRf>

DISPlay:NOISe:TEMPerature:X:START?

## Arguments

<NRf> is the Start Frequency value.

## Returns

<NR3> = floating point value with an exponent, which is the Start Frequency value.

## Examples

DISPLAY:NOISE:TEMPERATURE:X:START 1.5 GHz sets the Start Frequency in the Noise Temperature display to 1.5000 GHz.

DISPLAY:NOISE:TEMPERATURE:X:START? might return 1.500000000E+9, indicating that the Start Frequency in the Noise Temperature display is 1.5000 GHz.

## DISPlay:NOISe:TEMPerature:X:STOP

Sets or queries the Stop Frequency value in the Noise Temperature display.

## Conditions

Measurement views: Noise Temperature

## Group

Display commands

## Syntax

```
DISPlay:NOISe:TEMPerature:X:STOP <NRf>
```

```
DISPlay:NOISe:TEMPerature:X:STOP?
```

## Arguments

<NRf> is the Stop Frequency value.

## Returns

<NR3> = floating point value with an exponent, which is the Stop Frequency value.

## Examples

DISPLAY:NOISE:TEMPERATURE:X:STOP 1.5 GHz sets the Stop Frequency in the Noise Temperature display to 1.5000 GHz.

DISPLAY:NOISE:TEMPERATURE:X:STOP? might return 1.5000000000E+9, indicating that the Stop Frequency in the Noise Temperature display is 1.5000 GHz.

## DISPlay:NOISe:TEMPerature:Y:AUTO (No Query Form)

Automatically sets the vertical scale of the Noise Temperature graph.

## Conditions

Measurement view: Noise Temperature

## Group

Display commands

## Syntax

```
DISPlay:NOISe:TEMPerature:Y:AUTO
```

## Arguments

None

## DISPlay:NOISe:TEMPerature:Y:AUTO:STATe

Sets or queries the state of the automatic vertical scale and position of the graph to on or off.

## Conditions

Measurement view: Noise Temperature

## Group

Display commands

## Syntax

```
DISPlay:NOISe:TEMPerature:Y:AUTO:STATe { OFF | ON | 0 | 1 }
DISPlay:NOISe:TEMPerature:Y:AUTO:STATe?
```

## Arguments

OFF or 0 sets the analyzer to manual setting of the vertical scale and position of the graph.

ON or 1 sets the analyzer to automatically set the vertical scale and position of the graph.

## Returns

OFF or 0 means the analyzer is in manual mode for setting of the vertical scale and position of the graph.

ON or 1 means the analyzer is in automatic mode for setting of the vertical scale and position of the graph.

## Examples

DISPLAY:NOISE:TEMPERATURE:Y:AUTO:STATE ON sets the analyzer to automatically set the vertical scale and position of the graph.

## DISPlay:NOISe:TEMPerature:Y:PDIVision

Sets or queries the distance between graticule lines on the display in dB.

## Conditions

Measurement view: Noise Temperature

## Group

Display commands

## Syntax

```
DISPlay:NOISe:TEMPerature:Y:PDIVision <NRf>
DISPlay:NOISe:TEMPerature:Y:PDIVision?
```

## Arguments

<NRf> is the distance between graticule lines on the display.

Range: 100 mdB to 5.00 dB

## Returns

<NR2> = a floating point value without an exponent, which is the distance between graticule lines on the display in dB.

## Examples

DISPLAY:NOISE:TEMPERATURE:Y:PDIVISION? might return 1.500000000, indicating that the distance between graticule lines on the display is 1.50 dB.

## **DISPlay:NOISe:TEMPerature:Y:POSition**

Sets or queries the vertical position of the Noise Temperature display.

### **Conditions**

Measurement view: Noise Temperature

### **Group**

Display commands

### **Syntax**

```
DISPlay:NOISe:TEMPerature:Y:POSition <NRf>
```

```
DISPlay:NOISe:TEMPerature:Y:POSition?
```

### **Arguments**

<NRf> is the vertical position of the Noise Temperature display.

Range: 5.12 to 26.5

### **Returns**

<NR2> =a floating point value without an exponent, which is the vertical position value.

The vertical position value

### **Examples**

DISPLAY:NOISE:TEMPERATURE:Y:POSITION? might return 26.500000000, indicating that the vertical position is 26.5.

## **DISPlay:NOISe:TEMPerature:Y:SCALE**

Sets or queries the vertical scale of the Noise Temperature display.

### **Conditions**

Measurement view: Noise Temperature

### **Group**

Display commands

### **Syntax**

```
DISPlay:NOISe:TEMPerature:Y:SCALE <NRf>
```

```
DISPlay:NOISe:TEMPerature:Y:SCALE?
```

### **Arguments**

<NRf> is the vertical scale of the Noise Temperature display.

Range: 10.2 to 31.6

## Returns

<NR2> = a floating point value without an exponent, which is the vertical scale value.

## Examples

DISPLAY:NOISE:TEMPERATURE:Y:SCALE? might return 10.000000000, indicating that the vertical scale is 10.0 dB.

## DISPlay:NOISe:UNCERTainty:RESult:GAIN? (Query Only)

Queries the gain computed uncertainty value (dB).

## Conditions

Measurement view: Uncertainty Calculator

## Group

Display commands

## Syntax

DISPlay:NOISe:UNCERTainty:RESult:GAIN?

## Related Commands

[DISPlay:NOISe:UNCERTainty:RESult:NFIGure?](#)

## Returns

<NR3> = a floating point value with an exponent, which is the gain computed uncertainty value (dB).

## Examples

DISPLAY:NOISE:UNCERTAINTY:RESULT:GAIN? might return 783.0000000000E-3, indicating that the computed uncertainty value for gain is 0.783 dB.

## DISPlay:NOISe:UNCERTainty:RESult:NFIGure? (Query Only)

Queries the noise figure computed uncertainty value (dB).

## Conditions

Measurement view: Uncertainty Calculator

## Group

Display commands

## Syntax

DISPlay:NOISe:UNCERTainty:RESult:NFIGure?

## Related Commands

[\*DISPlay:NOISe:UNCERTainty:RESult:GAIN?\*](#)

## Returns

<NR3> = a floating point value with an exponent, which is the noise figure computed uncertainty value (dB).

## Examples

`DISPLAY:NOISE:UNCERTAINTY:RESULT:NFIGURE?` might return `783.0000000000E-3`, indicating that the computed uncertainty value for noise figure is 0.783 dB.

## DISPlay:NOISe:YFACTOR[:SCALE]:AUTO (No Query Form)

Automatically scales the Y Factor graph.

### Conditions

Measurement view: Y Factor

### Group

Display commands

### Syntax

```
DISPlay:NOISe:YFACTOR[:SCALE]:AUTO
```

### Arguments

None

## DISPlay:NOISe:YFACTOR:WINDOW:TRACE:GRATICULE:GRID:STATE

Sets or queries the graticule state in the Y Factor display to on (showing) or off (hidden).

### Conditions

Measurement view: Y Factor

### Group

Display commands

### Syntax

```
DISPlay:NOISe:YFACTOR:WINDOW:TRACE:GRATICULE:GRID:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:YFACTOR:WINDOW:TRACE:GRATICULE:GRID:STATE?
```

### Arguments

OFF or 0 hides the graticule in the Y Factor display.

ON or 1 shows the graticule in the Y Factor display.

## Returns

0 means the graticule in the Y Factor display is hidden (off).

1 means the graticule in the Y Factor display is showing (on).

## Examples

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:GRATICULE:GRID:STATE ON` will turn on the graticule in the Y Factor display.

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:GRATICULE:GRID:STATE?` might return 1, indicating that the graticule in the Y Factor display is showing (on).

## DISPlay:NOISe:YFActor:WINDow:TRACe:LEGend:STATe

Sets or queries the trace legend state in the Y Factor display to on (showing) or off (hidden).

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

`DISPlay:NOISe:YFActor:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }`

`DISPlay:NOISe:YFActor:WINDow:TRACe:LEGend:STATe?`

## Arguments

OFF or 0 hides the trace legend in the Y Factor display.

ON or 1 shows the trace legend in the Y Factor display.

## Returns

0 means the trace legend in the Y Factor display is hidden (off).

1 means the trace legend in the Y Factor display is showing (on).

## Examples

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:LEGEND:STATE ON` will turn on the trace legend in the Y Factor display.

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:LEGEND:STATE?1` indicates that the trace legend in the Y Factor display is showing (on).

## DISPlay:NOISe:YFActor:WINDow:TRACe:MEASPoints:STATe

Sets or queries the measurement points state in the Y Factor display to on (showing) or off (hidden).

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:WINDow:TRACe:MEASPoints:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:YFACTOR:WINDow:TRACe:MEASPoints:STATe?
```

## Arguments

OFF or 0 hides the measurement points in the Y Factor display.

ON or 1 shows the measurement points in the Y Factor display.

## Returns

0 means the measurement points in the Y Factor display is hidden (off).

1 means the measurement points in the Y Factor display is showing (on).

## Examples

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:MEASPOINTS:STATE ON` will turn on the measurement points in the Y Factor display.

`DISPLAY:NOISE:YFACTOR:WINDOW:TRACE:MEASPOINTS:STATE?` might return 1, indicating that the measurement points in the Y Factor display are showing (on).

## DISPlay:NOISe:YFACTOR:X:AUTO (No Query Form)

Automatically sets the horizontal scale of the Y Factor graph.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:X:AUTO
```

## Arguments

None

## DISPlay:NOISe:YFACTOR:X:OFFSet

Sets or queries the center frequency (offset) in the Y Factor display.



## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:X:OFFSet <NRf>
```

```
DISPlay:NOISe:YFACTOR:X:OFFSet?
```

## Arguments

<NRf> is the center frequency (offset) in the Y Factor display.

Range: -1.9900 GHz to +5.0000 GHz

## Returns

<NR3> = a floating point value with an exponent, which is the center frequency (offset) value.

## Examples

DISPLAY:NOISE:YFACTOR:X:OFFSET? might return 2.0000000009E+9, indicating that the center frequency (offset) value is 2.0000 GHz.

## DISPlay:NOISe:YFACTOR:X[:SCALE]

Sets or queries the horizontal scale of the Y Factor display.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:X[:SCALE] <NRf>
```

```
DISPlay:NOISe:YFACTOR:X[:SCALE]
```

## Related Commands

[DISPlay:NOISe:GAIN:Y:SCALE](#)

## Arguments

<NRf> is the horizontal scale of the Y Factor display.

Range: 100 Hz to 6.2 GHz

## Returns

<NR3> = a floating point value with an exponent, which is the horizontal scale value.

## Examples

DISP:NOISE:YFAC:X:SCAL? might return 2.990000000E+9, indicating that the horizontal scale value is 3.0 GHz.

## DISPlay:NOISe:YFACtor:X:STARt

Sets or queries the Start Frequency value in the Y Factor display.

## Conditions

Measurement views: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACtor:X:STARt <NRf>
```

```
DISPlay:NOISe:YFACtor:X:STARt?
```

## Arguments

<NRf> is the Start Frequency value in the Y Factor display.

## Returns

<NR3> = floating point value with an exponent, which is the Start Frequency value.

## Examples

DISPlay:NOISe:YFACtor:X:STARt 1.5 GHz sets the Start Frequency in the Y Factor display to 1.5000 GHz.

DISPlay:NOISe:YFACtor:X:STARt? might return 1.500000000E+9, indicating that the Start Frequency in the Y Factor display is 1.5000 GHz.

## DISPlay:NOISe:YFACtor:X:STOP

Sets or queries the Stop Frequency value Y Factor display.

## Conditions

Measurement views: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACtor:X:STOP <NRf>
```

```
DISPlay:NOISe:YFACTOR:X:STOP?
```

## Arguments

<NRf> is the Stop Frequency value in the Y Factor display.

## Returns

<NR3> = floating point value with an exponent, which is the Stop Frequency value.

## Examples

DISPLAY:NOISE:YFACTOR:X:STOP 1.5 GHz sets the Stop Frequency in the Y Factor display to 1.5000 GHz.

DISPLAY:NOISE:YFACTOR:X:STOP? might return 1.5000000000E+9, indicating that the Stop Frequency in the Y Factor display is 1.5000 GHz.

## DISPlay:NOISe:YFACTOR:Y:AUTO (No Query Form)

Automatically sets the vertical scale of the Y Factor graph.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:Y:AUTO
```

## Arguments

None

## DISPlay:NOISe:YFACTOR:Y:AUTO:STATe

Sets or queries the state of the automatic vertical scale and position of the Y Factor graph to on or off.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:Y:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:NOISe:YFACTOR:Y:AUTO:STATe?
```

## Arguments

OFF or 0 sets the analyzer to manual setting of the vertical scale and position of the graph.

ON or 1 sets the analyzer to automatically set the vertical scale and position of the graph.

## Returns

OFF or 0 means the analyzer is in manual mode for setting of the vertical scale and position of the graph.

ON or 1 means the analyzer is in automatic mode for setting of the vertical scale and position of the graph.

## Examples

`DISPLAY:NOISE:YFACTOR:Y:AUTO:STATE ON` sets the analyzer to automatically set the vertical scale and position of the graph.

## DISPlay:NOISe:YFACTOR:Y:PDIVision

Sets or queries the distance between graticule lines on the display in dB.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:Y:PDIVision <NRf>
```

```
DISPlay:NOISe:YFACTOR:Y:PDIVision?
```

## Arguments

<NRf> is the distance between graticule lines on the display in dB. Range: 100 mdB to 5.00 dB

## Returns

<NR2> = a floating point value without an exponent, which is the distance between graticule lines on the display in dB.

## Examples

`DISPLAY:NOISE:YFACTOR:Y:PDIVISION?` might return 1.5000000000, indicating that the distance between graticule lines on the display is 1.50 dB.

## DISPlay:NOISe:YFACTOR:Y:POSition

Sets or queries the vertical position of the Y Factor display.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:Y:POSition <NRf>
```

```
DISPlay:NOISe:YFACTOR:Y:POSition?
```

## Arguments

<NRf> is the vertical position of the Y Factor display.

Range: 5.12 to 26.5

## Returns

<NR2> = a floating point value without an exponent, which is the vertical position value.

## Examples

DISPLAY:NOISE:YFACTOR:Y:POSITION? might return 26.500000000, indicating that the vertical position is 26.5.

## DISPlay:NOISe:YFACTOR:Y:SCALE

Sets or queries the vertical scale of the Y Factor display.

## Conditions

Measurement view: Y Factor

## Group

Display commands

## Syntax

```
DISPlay:NOISe:YFACTOR:Y:SCALE <NRf>
```

```
DISPlay:NOISe:YFACTOR:Y:SCALE?
```

## Related Commands

## Arguments

<NRf> is the vertical scale of the Y Factor display.

Range: 10.2 to 31.6 dB

## Returns

<NR2> = a floating point value without an exponent, which is the vertical scale value.

## Examples

`DISPLAY:NOISE:YFACTOR:Y:SCALE?` might return 10.000000000, indicating that the vertical scale is 10.0 dB.

## DISPlay:NR:MEASview:DELeTe (No Query Form)

Deletes the corresponding 5G NR measurement view on the analysis window.

### Conditions

Measurement view: All 5G NR measurement view

### Group

Display commands

### Syntax

`DISPlay:NR:MEASview:DELeTe {CONS, SUMM, CHP, ACP, SEM, OBW, EVM, PVT}`

### Arguments

CONS deletes the 5G NR Constellation display.

SUMM deletes the 5G NR Summary display.

CHP deletes the 5G NR Channel Power display.

ACP deletes the 5G NR Adjacent Channel Power display.

SEM deletes the 5G NR Spectral Emission Mask display.

OBW deletes the 5G NR Occupied Bandwidth view.

EVM deletes the 5G NR EVM view.

PVT deletes the 5G NR Power vs Time view.

### Examples

`DISPLAY:NR:MEASVIEW:DELETE CONS`, will delete the 5G NR Constellation display.

## DISPlay:NR:MEASview:NEW (No Query Form)

Displays a new 5G NR measurement view on the analysis window.

### Conditions

Measurement view: All 5G NR measurement views

### Group

Display commands

### Syntax

`DISPlay:NR:MEASview:NEW {CONS, SUMM, CHP, ACP, SEM, OBW, EVM, PVT}`

## Arguments

CONS adds the 5G NR Constellation view.

SUMM adds the 5G NR Summary view.

CHP adds the 5G NR Channel Power view.

ACP adds the 5G NR Adjacent Channel Power view.

SEM adds the 5G NR Spectral Emission Mask view.

OBW adds the 5G NR Occupied Bandwidth view.

EVM adds the 5G NR EVM view.

PVT adds the 5G NR Power vs Time view.

## Examples

`DISPLAY:NR:MEASVIEW:NEW CONS`, will open the constellation display.

## DISPlay:NR:MEASview:SElect

Selects the corresponding 5G NR measurement view on the analysis window. The query command returns the currently selected view.

## Conditions

Measurement view: All 5G NR measurement views

## Group

Display commands

## Syntax

`DISPlay:NR:MEASview:SElect? (Query)`

`DISPlay:NR:MEASview:SElect {CONS, SUMM, CHP, ACP, SEM, OBW, EVM, PVT}`

## Arguments

CONS selects the 5G NR Constellation view.

SUMM selects the 5G NR Summary view.

CHP selects the 5G NR Channel Power view.

ACP selects the 5G NR Adjacent Channel Power view.

SEM selects the 5G NR Spectral Emission Mask view.

OBW selects the 5G NR Occupied Bandwidth view.

EVM selects the 5G NR EVM view.

PVT selects the 5G NR Power vs Time view.

## Examples

`DISPlay:NR:MEASview:SElect SUMM` selects Summary view.

## **DISPlay:OBWidth:MARKer:SHOW:STATe**

Determines whether to show or hide the readout for the selected marker in the Occupied Bandwidth view.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Display commands

### **Syntax**

```
DISPlay:OBWidth:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:OBWidth:MARKer:SHOW:STATe?
```

### **Arguments**

OFF or 0 hides the readout for the selected marker in the view.

ON or 1 shows the readout for the selected marker in the view.

### **Examples**

```
DISPLAY:OBWIDTH:MARKER:SHOW:STATE ON shows the readout for the selected marker in the view.
```

## **DISPlay:OBWidth:RESet:SCALE (No Query Form)**

Resets the horizontal and vertical scale to the default values described below in the Occupied Bandwidth view.

Vertical offset = Reference level, Vertical scale = 100 dB,

Horizontal offset = Center frequency, and Horizontal scale = Default span

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Display commands

### **Syntax**

```
DISPlay:OBWidth:RESet:SCALE
```

### **Arguments**

None

### **Examples**

```
DISPLAY:OBWIDTH:RESET:SCALE resets the horizontal and vertical scale to the default values.
```



## DISPlay:OBWidth:SElected:BANDwidth

Sets or queries the bandwidth (OBW or x dB BW) to measure in the Occupied Bandwidth view.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Display commands

### Syntax

```
DISPlay:OBWidth:SElected:BANDwidth { OBWidth | XDBbandwidth }  
DISPlay:OBWidth:SElected:BANDwidth?
```

### Arguments

OBWidth selects the occupied bandwidth to measure.

XDBbandwidth selects the x dB bandwidth to measure.

### Examples

```
DISPLAY:OBWIDTH:SELECTED:BANDWIDTHOBWidth selects the occupied bandwidth to measure.
```

## DISPlay:OBWidth:WINDow:TRACe:GRATICule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Display commands

### Syntax

```
DISPlay:OBWidth:WINDow:TRACe:GRATICule:GRID:STATe { OFF | ON | 0 | 1 }  
DISPlay:OBWidth:WINDow:TRACe:GRATICule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

### Examples

```
DISPLAY:OBWIDTH:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen.
```

## **DISPlay:OBWidth:X[:SCALE]**

Sets or queries the horizontal range of the Occupied Bandwidth view.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Display commands

### **Syntax**

```
DISPlay:OBWidth:X[:SCALE] <value>
```

```
DISPlay:OBWidth:X[:SCALE]?
```

### **Related Commands**

[\*DISPlay:OBWidth:X\[:SCALE\]:OFFSet\*](#)

### **Arguments**

<value>::<NRf> specifies the horizontal range.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

### **Examples**

DISPLAY:OBWIDTH:X:SCALE 10MHz sets the horizontal range to 10 MHz.

## **DISPlay:OBWidth:X[:SCALE]:AUTO (No Query Form)**

Rescales the horizontal axis automatically to fit the waveform to the screen in the Occupied Bandwidth view.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Display commands

### **Syntax**

```
DISPlay:OBWidth:X[:SCALE]:AUTO
```

### **Arguments**

None

## Examples

`DISPLAY:OBWIDTH:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:OBWidth:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Occupied Bandwidth view.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Display commands

### Syntax

`DISPlay:OBWidth:X[:SCALe]:OFFSet <value>`

`DISPlay:OBWidth:X[:SCALe]:OFFSet?`

### Related Commands

[DISPlay:OBWidth:X\[:SCALe\]](#)

### Arguments

`<value>::<NRf>` specifies the minimum horizontal value.

Range:  $[(\text{center frequency}) - (X \text{ scale}) \times 0.9]$  to  $[(\text{center frequency}) + (X \text{ scale}) \times 0.9]$

### Examples

`DISPLAY:OBWIDTH:X:SCALE:OFFSET 1.45GHz` sets the minimum horizontal value to 1.45 GHz in the Occupied Bandwidth view.

## DISPlay:OBWidth:Y[:SCALe]

Sets or queries the vertical range of the Occupied Bandwidth view.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Display commands

### Syntax

`DISPlay:OBWidth:Y[:SCALe] <value>`

`DISPlay:OBWidth:Y[:SCALe]?`

## Related Commands

[\*DISPlay:OBWidth:Y\[:SCALE\]:OFFSet\*](#)

## Arguments

<value>::<Nrf> specifies the vertical range. Range: 0.1 to 200 dB.

## Examples

DISPLAY:OBWIDTH:Y:SCALE 100 sets the vertical range to 100 dB in the Occupied Bandwidth view.

## DISPlay:OBWidth:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Occupied Bandwidth view.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Display commands

## Syntax

DISPlay:OBWidth:Y[:SCALE]:AUTO

## Arguments

None

## Examples

DISPLAY:OBWIDTH:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:OBWidth:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the Occupied Bandwidth view.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Display commands

## Syntax

DISPlay:OBWidth:Y[:SCALE]:OFFSet <value>

DISPlay:OBWidth:Y[:SCALE]:OFFSet?

## Related Commands

[DISPlay:OBWidth:Y\[:SCALE\]](#)

## Arguments

<value>::<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

## Examples

DISPLAY:OBWIDTH:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBm in the Occupied Bandwidth view.

## DISPlay:OFDM:CONSte[:SCALE]

Sets or queries the scale (zoom) in the OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CONSte[:SCALE] <value>
```

```
DISPlay:OFDM:CONSte[:SCALE]?
```

## Arguments

<value>::<NRf> specifies the zoom setting in the OFDM Constellation view.

A positive value zooms out and a negative value zooms in.

## Examples

DISPLAY:OFDM:CONSTE[:SCALE] 5 zooms the display out by 5

## DISPlay:OFDM:CONSte:X:OFFSet

Sets or queries the horizontal offset in the OFDM Constellation view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CONSte:X:OFFSet <value>
```

DISPlay:OFDM:CONStE:X:OFFSet?

### Arguments

<value> : : <NRf> specifies the offset range.

### Examples

DISPLAY:OFDM:CONSTE:X:OFFSET 2.7 sets the horizontal offset to 2.7 units to the left.

## DISPlay:OFDM:CONStE:Y:OFFSet

Sets or queries the vertical offset in the OFDM Constellation view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:CONStE:Y:OFFSet <value>

DISPlay:OFDM:CONStE:Y:OFFSet?

### Arguments

<value> : : <NRf> specifies the offset range.

A positive value moves the offset up; and negative value moves the offset down.

### Examples

DISPLAY:OFDM:CONSTE:Y:OFFSET 2 sets the vertical offset to up 2 units.

## DISPlay:OFDM:CRESpOse:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:CRESpOse:FREQuency:AUTO

## Arguments

None

## Examples

`DISPLAY:OFDM:CRESPONSE:FREQUENCY:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:CRESpone:FREQuency:OFFSet

Sets or queries the frequency offset in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:CRESpone:FREQuency:OFFSet <value>`

`DISPlay:OFDM:CRESpone:FREQuency:OFFSet?`

### Arguments

`<value>::<NRf>` specifies the frequency offset.

### Examples

`DISPLAY:OFDM:CRESPONSE:FREQUENCY:OFFSET 1.45E+9` sets the frequency offset to 1.45 GHz.

## DISPlay:OFDM:CRESpone:FREQuency[:SCALe]

Sets or queries the horizontal range of the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:CRESpone:FREQuency[:SCALe] <value>`

`DISPlay:OFDM:CRESpone:FREQuency[:SCALe]?`

### Arguments

`<value>::= <NRf>` specifies the horizontal range.

## Examples

`DISPLAY:OFDM:CRESPONSE:FREQUENCY[:SCALE] 10E+6` sets the horizontal range to 10 MHz.

## DISPlay:OFDM:CRESpOse:MAGNitude:AUTO (No Query Form)

Rescales the magnitude automatically to fit the waveform to the screen in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:CRESpOse:MAGNitude:AUTO`

### Arguments

None

## Examples

`DISPLAY:OFDM:CRESPONSE:MAGNITUDE:AUTO` rescales the magnitude scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:CRESpOse:MAGNitude:OFFSet

Sets or queries the magnitude offset in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:CRESpOse:MAGNitude:OFFSet <value>`

`DISPlay:OFDM:CRESpOse:MAGNitude:OFFSet?`

### Arguments

`<value> ::= <NRf>` specifies the magnitude offset.

## Examples

`DISPLAY:OFDM:CRESPONSE:MAGNITUDE:OFFSET 5` sets the magnitude offset to 5 dB.



---

## DISPlay:OFDM:CRESpOse:MAGNitude:PDIVision

Sets or queries the scale (per division) of the magnitude in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:CRESpOse:MAGNitude:PDIVision <value>
```

```
DISPlay:OFDM:CRESpOse:MAGNitude:PDIVision?
```

### Arguments

<value>::<Nrf> specifies the vertical scale (per division).

### Examples

```
DISPLAY:OFDM:CRESPONSE:MAGNITUDE:PDIVISION 2.7 sets the vertical scale to 2.70 dB/div.
```

## DISPlay:OFDM:CRESpOse:PHASe:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the OFDM Channel response graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:CRESpOse:PHASe:AUTO
```

### Arguments

None

### Examples

```
DISPLAY:OFDM:CRESPONSE:PHASE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.
```

## DISPlay:OFDM:CRESpOse:PHASe:OFFSet

Sets or queries the phase offset in the OFDM Channel response graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CRESpOse:PHASe:OFFSet <value>
```

```
DISPlay:OFDM:CRESpOse:PHASe:OFFSet?
```

## Arguments

<value>::<NRf> specifies the phase offset.

## Examples

DISPLAY:OFDM:CRESPONSE:PHASE:OFFSET -14.5 sets the minimum vertical value to -14.5 ° in the OFDM Channel response graph.

## DISPlay:OFDM:CRESpOse:PHASe[:SCALE]

Sets the vertical scale automatically to fit the waveform to the screen in the OFDM Channel response graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CRESpOse:PHASe[:SCALE] <value>
```

```
DISPlay:OFDM:CRESpOse:PHASe[:SCALE]?
```

## Arguments

<value>::<NRf> specifies the vertical range.

## Examples

DISPLAY:OFDM:CRESPONSE:PHASE[:SCALE] 30 sets the vertical range to 30 ° in the OFDM Channel response graph.

## DISPlay:OFDM:CRESpOse:WINDow:SELEct:PLOT

Sets or queries the plot in the OFDM Channel Response window measurement.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CRESpOse:WINDow:SElect:PLOT { MAGNitude | PHASe }
```

```
DISPlay:OFDM:CRESpOse:WINDow:SElect:PLOT?
```

## Arguments

MAGNitude selects the magnitude display.

PHASe selects the phase display

## Examples

```
DISPLAY:OFDM:CRESPONSE:WINDOW:SELECT:PLOT PHASe selects the phase display.
```

## DISPlay:OFDM:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the OFDM Channel Response view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }
```

```
DISPlay:OFDM:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the graph.

ON or 1 shows the graticule in the graph.

## Examples

```
DISPLAY:OFDM:CRESPONSE:WINDOW:TRACE:GRATICULE:GRID:STATE OFF turns off the graticule in the graph.
```

## DISPlay:OFDM:EVM:FREQuency:AUTO (No Query Form)

Rescales the frequency automatically to fit the waveform to the screen in the OFDM EVM graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:EVM:FREQuency:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:EVM:FREQUENCY:AUTO rescales the frequency scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:EVM:FREQuency:OFFSet

Sets or queries the frequency offset in the OFDM EVM graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:EVM:FREQuency:OFFSet <value>
```

```
DISPlay:OFDM:EVM:FREQuency:OFFSet
```

## Arguments

<value>::<NRf> specifies the frequency offset.

## Examples

DISPLAY:OFDM:EVM:FREQUENCY:OFFSET -21 sets the frequency offset to -21 subcarriers.

## DISPlay:OFDM:EVM:FREQuency[:SCALE]

Sets or queries the horizontal range of the OFDM EVM graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:EVM:FREQuency[:SCALE] <value>
```

```
DISPlay:OFDM:EVM:FREQuency[:SCALE]?
```

## Arguments

<value> ::= <Nrf> specifies the horizontal range.

## Examples

```
DISPLAY:OFDM:EVM:FREQUENCY[:SCALE] 10E+6 sets the horizontal range to 10 MHz.
```

## DISPlay:OFDM:EVM:MARKer:SHOW:STATE

Determines whether to show or hide the readout for the selected marker in the OFDM EVM view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:EVM:MARKer:SHOW:STATE { OFF | ON | 1 | 0 }
```

```
DISPlay:OFDM:EVM:MARKer:SHOW:STATE?
```

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

```
DISPLAY:OFDM:EVM:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.
```

## DISPlay:OFDM:EVM:TIME:AUTO (No Query Form)

Rescales the time automatically to fit the waveform to the screen in the OFDM EVM view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:EVM:TIME:AUTO
```

## Arguments

None

## Examples

`DISPLAY:OFDM:EVM:TIME:AUTO` rescales the time scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:EVM:TIME:OFFSet

Sets or queries the time offset in the OFDM EVM view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:EVM:TIME:OFFSet <value>`

`DISPlay:OFDM:EVM:TIME:OFFSet?`

## Arguments

`<value> ::= <NRf>` specifies the magnitude offset.

## Examples

`DISPLAY:OFDM:EVM:TIME:OFFSET -11.7` sets the time scale offset to  $-11.7$  symbols.

## DISPlay:OFDM:EVM:TIME[:SCALE]

Sets or queries the time scale in the OFDM EVM view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:EVM:TIME[:SCALE] <value>`

`DISPlay:OFDM:EVM:TIME[:SCALE]?`

## Arguments

`<value> ::= <NRf>` specifies the time scale.

## Examples

`DISPLAY:OFDM:EVM:TIME[:SCALE] 13` sets the time scale to 13 symbols.

## DISPlay:OFDM:EVM:WINDow:SElect:PLOT

Sets or queries the plot in the OFDM EVM window measurement.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:EVM:WINDow:SElect:PLOT { BOTH | SCARrier | SYMBols }  
DISPlay:OFDM:EVM:WINDow:SElect:PLOT?
```

### Arguments

`SCARrier` selects the subcarrier display.

`SYMBols` selects the symbols display

`BOTH` selects both displays.

## Examples

`DISPLAY:OFDM:EVM:WINDOW:SELECT:PLOT SYMBols` selects the symbols display.

## DISPlay:OFDM:EVM:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the OFDM EVM view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:EVM:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }  
DISPlay:OFDM:EVM:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

`OFF` or `0` hides the graticule in the graph.

`ON` or `1` shows the graticule in the graph.

## Examples

`DISPLAY:OFDM:EVM:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` turns off the graticule in the graph.

## DISPlay:OFDM:EVM:Y:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform in the OFDM EVM view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:EVM:Y:AUTO`

### Arguments

None

## Examples

`DISPLAY:OFDM:EVM:Y:AUTO` rescales the vertical scale to fit the waveform on the screen.

## DISPlay:OFDM:EVM:Y:OFFSet

Sets or queries the vertical offset in the OFDM EVM view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:EVM:Y:OFFSet <value>`

`DISPlay:OFDM:EVM:Y:OFFSet?`

### Arguments

`<value> ::= <NRf>` specifies the vertical offset.

## Examples

`DISPLAY:OFDM:EVM:Y:OFFSET -20` sets the vertical offset to -20%.



## DISPlay:OFDM:EVM:Y[:SCALE]

Sets or queries the vertical range of the OFDM EVM view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:EVM:Y[:SCALE] <value>

DISPlay:OFDM:EVM:Y[:SCALE]?

### Arguments

<value>::= <NRf> specifies the vertical scale.

### Examples

DISPLAY:OFDM:EVM:Y[:SCALE] 200 sets the vertical scale to 200%.

## DISPlay:OFDM:FLATness:AUTO (No Query Form)

Automatically rescales the horizontal and vertical axes for the best display in the OFDM Spectral Flatness view.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Display commands

### Syntax

DISPlay:OFDM:FLATness:AUTO

### Arguments

None

### Examples

DISP:OFDM:FLAT:AUTO automatically rescales the horizontal and vertical axes for the best display.

## DISPlay:OFDM:FLATness:X:AUTO (No Query Form)

Rescales the horizontal axis automatically in the OFDM Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Display commands

## Syntax

```
DISPlay:OFDM:FLATness:X:AUTO
```

## Arguments

None

## Examples

`DISP:OFDM:FLAT:X:AUTO` rescales the horizontal axis automatically for the best display.

## DISPlay:OFDM:FLATness:X:OFFSet

Sets or queries the value of the offset of the horizontal axis from the center of the OFDM WLAN Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Display commands

## Syntax

```
DISPlay:OFDM:FLATness:X:OFFSet <NRf>
```

```
DISPlay:OFDM:FLATness:X:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 recenters the plot.

## Returns

The value of the offset of the X axis from the center of the plot as a floating point number.

## Examples

`DISP:OFDM:FLAT:X:OFFS 150` shifts the display to the left by 150%.

## DISPlay:OFDM:FLATness:X[:SCALE]

Sets or queries the value of the horizontal scale in the OFDM Spectral Flatness display. The units are Frequency or Subcarrier. To set the units, use the command [SENSe]:OFDM:UNIT:FREQuency.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Display commands

### Syntax

```
DISPlay:OFDM:FLATness:X[:SCALE] <Nrf>
```

```
DISPlay:OFDM:FLATness:X[:SCALE]?
```

### Arguments

Floating point number that represents the value of the horizontal scale.

### Examples

```
DISP:OFDM:FLAT:X:SCALE 32 sets the horizontal scale to 32.
```

## DISPlay:OFDM:FLATness:Y:AUTO (No Query Form)

Rescales the vertical axis scale and position values automatically in the OFDM Spectral Flatness display.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Display commands

### Syntax

```
DISPlay:OFDM:FLATness:Y:AUTO
```

### Arguments

None

### Examples

```
DISP:OFDM:FLAT:Y:AUTO automatically selects the vertical scale and position values.
```

## DISPlay:OFDM:FLATness:Y:OFFSet

Sets or queries the value of the offset from the center (vertical position), in percent terms, for the OFDM Spectral Flatness display. To set the Vertical Scale, use the command DISPlay:OFDM:FLATness:Y[:SCALE]

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Display commands

### Syntax

```
DISPlay:OFDM:FLATness:Y:OFFSet <NRf>
```

```
DISPlay:OFDM:FLATness:Y:OFFSet?
```

### Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

### Returns

The value of the offset from the center of the Y axis for the plot as a floating point number.

### Examples

```
DISP:OFDM:FLAT:Y:OFFS 150 sets the offset from the center (vertical position) of the plot to 150%.
```

## DISPlay:OFDM:FLATness:Y[:SCALE]

Sets or queries the vertical scale value for the OFDM Spectral Flatness display. To set the vertical position (offset) value, use the command DISPlay:OFDM:FLATness:Y:OFFSet.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Display commands

### Syntax

```
DISPlay:OFDM:FLATness:Y[:SCALE] <NRf>
```

```
DISPlay:OFDM:FLATness:Y[:SCALE]?
```

## Arguments

Floating point number that specifies the vertical scale value.

## Examples

`DISP:OFDM:FLAT:Y:SCALE 10` sets the vertical scale value to 10.

## DISPlay:OFDM:MEASview:DELeTe (No Query Form)

Deletes the specified OFDM view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:MEASview:DELeTe { CONSTe | SUMMary | STABle | CRESpone | EVM | PERRor
| MERRor | POWer }
```

## Arguments

CONSte deletes the OFDM Constellation view.

SUMMary deletes the OFDM Summary view.

STABle deletes the OFDM Symbol table view.

CRESpone deletes the OFDM Channel response view.

EVM deletes the EVM (Error Vector Magnitude) versus Time view.

PERRor deletes the Phase error versus Time view.

MERRor deletes the Magnitude error versus Time view.

POWer deletes the OFDM Power view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

`DISPLAY:OFDM:MEASVIEW:DELETE CONSTe` deletes the OFDM Constellation view.

## DISPlay:OFDM:MEASview:NEW (No Query Form)

Displays a new OFDM measurement view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:MEASview:NEW { CONSte | SUMMary | STABle | CRESpone | EVM | PERRor |  
MERRor | POWer }
```

## Arguments

CONSte creates a new OFDM Constellation view.

SUMMary creates a new OFDM Summary view.

STABle creates a new OFDM Symbol table view.

CRESpone creates a new OFDM Channel response view.

EVM creates a new EVM (Error Vector Magnitude) versus Time view.

PERRor creates a new Phase error versus Time view.

MERRor creates a new Magnitude error versus Time view.

POWer creates a new OFDM Power view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

```
DISPLAY:OFDM:MEASVIEW:NEW:CONSte creates a new OFDM Constellation view.
```

## DISPlay:OFDM:MEASview:SElect

Selects a OFDM measurement view on the screen. The query returns the currently selected view.

Selecting a measurement optimizes it. Other measurements may be optimized as a side effect. Refer to the DISPlay:WINDow:OPTimized:MEASurement? query.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:MEASview:SElect { CONSte | SUMMary | STABle | CRESpone | EVM | PERRor  
| MERRor | POWer }
```

```
DISPlay:OFDM:MEASview:SElect?
```

## Arguments

CONSte creates a new OFDM Constellation view.

SUMMary creates a new OFDM Summary view.

STABLE creates a new OFDM Symbol table view.

CRESPonse creates a new OFDM Channel response view.

EVM creates a new EVM (Error Vector Magnitude) versus Time view.

PERRor creates a new Phase error versus Time view.

MERRor creates a new Magnitude error versus Time view.

POWer creates a new OFDM Power view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

DISPLAY:OFDM:MEASVIEW:SELECT:STABLE selects the OFDM symbol table view.

## DISPlay:OFDM:MERRor:FREQuency[:SCALE]

Sets or queries the horizontal range of the OFDM Magnitude error graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:MERRor:FREQuency[:SCALE] <value>

DISPlay:OFDM:MERRor:FREQuency[:SCALE]?

### Arguments

<value>::= <NRf> specifies the horizontal range.

## Examples

DISPLAY:OFDM:MERROR:FREQUENCY[:SCALE] 10E+6 sets the horizontal range to 10 MHz.

## DISPlay:OFDM:MERRor:FREQuency[:SCALE]:AUTO (No Query Form)

Rescales the frequency automatically to fit the waveform to the screen in the OFDM Magnitude error graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

## Syntax

```
DISPlay:OFDM:MERRor:FREQuency[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:MERROR:FREQUENCY[:SCALE]:AUTO rescales the frequency scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:MERRor:FREQuency[:SCALE]:OFFSet

Sets or queries the frequency offset in the OFDM Magnitude error graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:MERRor:FREQuency[:SCALE]:OFFSet <value>  
DISPlay:OFDM:MERRor:FREQuency[:SCALE]:OFFSet?
```

## Arguments

<value>::<NRf> specifies the frequency offset.

## Examples

DISPLAY:OFDM:MERROR:FREQUENCY[:SCALE]:OFFSET -26 sets the offset to -26 subcarriers.

## DISPlay:OFDM:MERRor:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the OFDM Magnitude error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:MERRor:MARKer:SHOW:STATe { OFF | ON | 1 | 0 }  
DISPlay:OFDM:MERRor:MARKer:SHOW:STATe?
```



## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

`DISPLAY:OFDM:MERROR:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

## DISPlay:OFDM:MERRor:TIME[:SCALE]

Sets or queries the time scale in the OFDM Magnitude error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:MERRor:TIME[:SCALE] <value>`

`DISPlay:OFDM:MERRor:TIME[:SCALE]?`

## Arguments

`<value>::= <NRf>` specifies the time scale.

## Examples

`DISPLAY:OFDM:MERROR:TIME[:SCALE] 50` sets the time scale to 50 subcarriers.

## DISPlay:OFDM:MERRor:TIME[:SCALE]:AUTO (No Query Form)

Rescales the time automatically to fit the waveform to the screen in the OFDM Magnitude error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:MERRor:TIME[:SCALE]:AUTO`

## Arguments

None

## Examples

`DISPLAY:OFDM:MERROR:TIME[:SCALE]:AUTO` rescales the time scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:MERRor:TIME[:SCALE]:OFFSet

Sets or queries the time offset in the OFDM Magnitude error view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:MERRor:TIME[:SCALE]:OFFSet <value>`

`DISPlay:OFDM:MERRor:TIME[:SCALE]:OFFSet?`

### Arguments

`<value> ::= <NRf>` specifies the time offset.

## Examples

`DISPLAY:OFDM:MERROR:TIME[:SCALE]:OFFSET -47` sets the offset to `-47` symbols.

## DISPlay:OFDM:MERRor:WINDow:SElect:PLOT

Sets or queries the plot in the OFDM Magnitude error window measurement.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:MERRor:WINDow:SElect:PLOT { BOTH | SCARrier | SYMBols }`

`DISPlay:OFDM:MERRor:WINDow:SElect:PLOT?`

### Arguments

`SCARrier` selects the subcarrier display.

`SYMBols` selects the symbols display

`BOTH` selects both displays.

## Examples

`DISPLAY:OFDM:MERROR:WINDOW:SELECT:PLOT SYMBOLS` selects the symbols display.

## DISPlay:OFDM:MERRor:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the OFDM Magnitude error view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:MERRor:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }`

`DISPlay:OFDM:MERRor:WINDow:TRACe:GRATicule:GRID:STATe?`

### Arguments

OFF or 0 hides the graticule in the graph.

ON or 1 shows the graticule in the graph.

## Examples

`DISPLAY:OFDM:MERROR:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` turns off the graticule in the graph.

## DISPlay:OFDM:MERRor:Y:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform in the OFDM Magnitude error view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

`DISPlay:OFDM:MERRor:Y:AUTO`

### Arguments

None

## Examples

`DISPLAY:OFDM:MERROR:Y:AUTO` rescales the vertical scale to fit the waveform on the screen.

## **DISPlay:OFDM:MERRor:Y:OFFSet**

Sets or queries the vertical offset in the OFDM Magnitude error view.

### **Conditions**

Measurement views: OFDM

### **Group**

Display commands

### **Syntax**

```
DISPlay:OFDM:MERRor:Y:OFFSet <value>
```

```
DISPlay:OFDM:MERRor:Y:OFFSet?
```

### **Arguments**

<value> ::= <NRf> specifies the vertical offset.

### **Examples**

```
DISPLAY:OFDM:MERROR:Y:OFFSET -90 sets the vertical scale offset to -90%.
```

## **DISPlay:OFDM:MERRor:Y[:SCALE]**

Sets or queries the vertical range of the OFDM Magnitude error view.

### **Conditions**

Measurement views: OFDM

### **Group**

Display commands

### **Syntax**

```
DISPlay:OFDM:MERRor:Y[:SCALE] <value>
```

```
DISPlay:OFDM:MERRor:Y[:SCALE]?
```

### **Arguments**

<value> ::= <NRf> specifies the vertical scale.

### **Examples**

```
DISPLAY:OFDM:MERROR:Y[:SCALE] 100 sets the vertical scale to 100%.
```

## **DISPlay:OFDM:PERRor:FREQuency[:SCALE]**

Sets or queries the horizontal range of the OFDM Phase error graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:FREQuency[:SCALe] <value>
```

```
DISPlay:OFDM:PERRor:FREQuency[:SCALe]?
```

## Arguments

<value> ::= <NRf> specifies the horizontal range.

## Examples

DISPLAY:OFDM:PERROR:FREQUENCY[:SCALE] 10E+6 sets the horizontal range to 10 MHz.

## DISPlay:OFDM:PERRor:FREQuency[:SCALe]:AUTO (No Query Form)

Rescales the frequency automatically to fit the waveform to the screen in the OFDM Phase error graph.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:FREQuency[:SCALe]:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:PERROR:FREQUENCY[:SCALE]:AUTO rescales the frequency scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:PERRor:FREQuency[:SCALe]:OFFSet

Sets or queries the frequency offset in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:FREQUency[:SCALE]:OFFSet <value>
```

```
DISPlay:OFDM:PERRor:FREQUency[:SCALE]:OFFSet?
```

## Arguments

<value> ::= <Nrf> specifies the frequency offset.

## Examples

```
DISPLAY:OFDM:ERROR:FREQUENCY[:SCALE]:OFFSET -26 sets the frequency scale offset to -26 subcarriers.
```

## DISPlay:OFDM:PERRor:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:MARKer:SHOW:STATe { OFF | ON | 1 | 0 }
```

```
DISPlay:OFDM:PERRor:MARKer:SHOW:STATe?
```

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

```
DISPLAY:OFDM:ERROR:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.
```

## DISPlay:OFDM:PERRor:TIME[:SCALE]

Sets or queries the time scale in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:TIME[:SCALe] <value>
```

```
DISPlay:OFDM:PERRor:TIME[:SCALe]?
```

## Arguments

<value>::= <Nrf> specifies the time scale.

## Examples

DISPLAY:OFDM:ERROR:TIME[:SCALE] 50 sets the time scale to 50 symbols.

## DISPlay:OFDM:PERRor:TIME[:SCALe]:AUTO (No Query Form)

Rescales the time automatically to fit the waveform to the screen in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:TIME[:SCALe]:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:ERROR:TIME[:SCALE]:AUTO rescales the time scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:PERRor:TIME[:SCALe]:OFFSet

Sets or queries the time offset in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:PERRor:TIME[:SCALe]:OFFSet <value>
```

```
DISPlay:OFDM:PERRor:TIME[:SCALe]:OFFSet?
```

## Arguments

<value> ::= <NRf> specifies the time offset.

## Examples

DISPLAY:OFDM:ERROR:TIME[:SCALE]:OFFSET -17 sets the time scale offset to -17 symbols.

## DISPlay:OFDM:PERRor:WINDow:SElect:PLOT

Sets or queries the plot in the OFDM Phase error window measurement.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

DISPlay:OFDM:PERRor:WINDow:SElect:PLOT { BOTH | SCARrier | SYMBols }

DISPlay:OFDM:PERRor:WINDow:SElect:PLOT?

## Arguments

SCARrier selects the subcarrier display.

SYMBols selects the symbols display

BOTH selects both displays.

## Examples

DISPLAY:OFDM:ERROR:WINDOW:SELECT:PLOT SYMBols selects the symbols display.

## DISPlay:OFDM:PERRor:WINDow:TRACe:GRATicule:GRID:STATE

Determines whether to show or hide the graticule in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

DISPlay:OFDM:PERRor:WINDow:TRACe:GRATicule:GRID:STATE { OFF | ON | 1 | 0 }

DISPlay:OFDM:PERRor:WINDow:TRACe:GRATicule:GRID:STATE?



## Arguments

OFF or 0 hides the graticule in the graph.

ON or 1 shows the graticule in the graph.

## Examples

`DISPLAY:OFDM:PERROR:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` turns off the graticule in the graph.

## DISPlay:OFDM:PERRor:Y:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:PERRor:Y:AUTO`

## Arguments

None

## Examples

`DISPLAY:OFDM:PERROR:Y:AUTO` rescales the vertical scale to fit the waveform on the screen.

## DISPlay:OFDM:PERRor:Y:OFFSet

Sets or queries the vertical offset in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

`DISPlay:OFDM:PERRor:Y:OFFSet <value>`

`DISPlay:OFDM:PERRor:Y:OFFSet?`

## Arguments

`<value>::= <NRf>` specifies the horizontal range.

## Examples

DISPLAY:OFDM:PERROR:Y:OFFSET -173 sets the vertical offset to -173 °.

## DISPlay:OFDM:PERRor:Y[:SCALE]

Sets or queries the vertical range of the OFDM Phase error view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:PERRor:Y[:SCALE] <value>

DISPlay:OFDM:PERRor:Y[:SCALE]?

### Arguments

<value>::= <Nrf> specifies the vertical scale.

## Examples

DISPLAY:OFDM:PERROR:Y[:SCALE] 200 sets the vertical scale to 200 °.

## DISPlay:OFDM:POWer:FREQuency[:AUTO](No Query Form)

Rescales the frequency automatically to fit the waveform to the screen in the OFDM Power graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

DISPlay:OFDM:POWer:FREQuency[:AUTO]

### Arguments

None

## Examples

DISPLAY:OFDM:POWER:FREQUENCY[:AUTO] rescales the frequency scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:POWer:FREQuency[:OFFSet]

Sets or queries the frequency offset in the OFDM Power view.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:POWer:FREQuency[:OFFSet] <value>
```

```
DISPlay:OFDM:POWer:FREQuency[:OFFSet]?
```

### Arguments

<value> ::= <NRf> specifies the frequency offset.

### Examples

```
DISPLAY:OFDM:POWER:FREQUENCY[:OFFSET] -26 sets the frequency offset to -26 subcarriers.
```

## DISPlay:OFDM:POWer:FREQuency[:SCALe]

Sets or queries the horizontal range of the OFDM Power graph.

### Conditions

Measurement views: OFDM

### Group

Display commands

### Syntax

```
DISPlay:OFDM:POWer:FREQuency[:SCALe] <value>
```

```
DISPlay:OFDM:POWer:FREQuency[:SCALe]?
```

### Arguments

<value> ::= <NRf> specifies the horizontal range.

### Examples

```
DISPLAY:OFDM:POWER:FREQUENCY[:SCALE] 10E+6 sets the horizontal range to 10 MHz.
```

## DISPlay:OFDM:POWer:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:MARKer:SHOW:STATe { OFF | ON | 1 | 0 }
```

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

DISPLAY:OFDM:POWER:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.

## DISPlay:OFDM:POWer:TIME[:AUTO (No Query Form)

Rescales the time automatically to fit the waveform to the screen in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:TIME[:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:POWER:TIME[:AUTO rescales the time scale automatically to fit the waveform to the screen.

## DISPlay:OFDM:POWer:TIME:OFFSet

Sets or queries the time offset in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:TIME:OFFSet <value>
```

```
DISPlay:OFDM:POWer:TIME:OFFSet?
```

## Arguments

<value> ::= <NR1> specifies the time offset.

## Examples

DISPLAY:OFDM:POWER:TIME:OFFSET? might return 2.00000 indicating the offset is 2 symbols.

## DISPlay:OFDM:POWer:TIME[:SCALE]

Sets or queries the time scale in the OFDM Phase error view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:TIME[:SCALE] <value>
```

```
DISPlay:OFDM:POWer:TIME[:SCALE]?
```

## Arguments

<value> ::= <NRf> specifies the time scale.

## Examples

DISPLAY:OFDM:POWER:TIME[:SCALE] 46 sets the time scale to 46 symbols.

## DISPlay:OFDM:POWer:WINDow:SElect:PLOT

Sets or queries the plot in the OFDM Power window measurement.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:WINDow:SElect:PLOT { BOTH | SCARrier | SYMBols }  
DISPlay:OFDM:POWer:WINDow:SElect:PLOT?
```

## Arguments

SCARrier selects the subcarrier display.

SYMBols selects the symbols display

BOTH selects both displays.

## Examples

```
DISPLAY:OFDM:POWER:WINDOW:SELECT:PLOT SYMBols selects the symbols display.
```

## DISPlay:OFDM:POWer:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }  
DISPlay:OFDM:POWer:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the graph.

ON or 1 shows the graticule in the graph.

## Examples

```
DISPLAY:OFDM:POWER:WINDOW:TRACE:GRATICULE:GRID:STATE OFF turns off the graticule in the graph.
```

## DISPlay:OFDM:POWer:Y:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:Y:AUTO
```

## Arguments

None

## Examples

DISPLAY:OFDM:POWER:Y:AUTO rescales the vertical scale to fit the waveform on the screen.

## DISPlay:OFDM:POWer:Y:OFFSet

Sets or queries the vertical offset in the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:Y:OFFSet <value>
```

```
DISPlay:OFDM:POWer:Y:OFFSet?
```

## Arguments

<value> ::= <NRf> specifies the horizontal range.

## Examples

DISPLAY:OFDM:POWER:Y:OFFSET -37.29 sets the vertical offset to -37.29 dBm.

## DISPlay:OFDM:POWer:Y[:SCALE]

Sets or queries the vertical range of the OFDM Power view.

## Conditions

Measurement views: OFDM

## Group

Display commands

## Syntax

```
DISPlay:OFDM:POWer:Y[:SCALE] <value>
```

```
DISPlay:OFDM:POWer:Y[:SCALE]?
```

## Arguments

<value> ::= <NRf> specifies the vertical scale.

## Examples

DISPLAY:OFDM:POWER:Y[:SCALE] 64.48 sets the vertical scale to 64.48 dB.

## DISPlay:P25:CONStE:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries to show or hide the graticule grid on the screen in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation

## Group

Display commands

## Syntax

```
DISPlay:P25:CONStE:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:P25:CONStE:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

0 means that the graticule grid is hidden.

1 means that the graticule grid is showing.

## Examples

```
DISPLAY:P25:CONSTE:WINDOW:TRACE:GRATICULE:GRID:STATE ON
```

shows the graticule grid on the screen.

## DISPlay:P25:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen in the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Display commands



## Syntax

```
DISPlay:P25:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe {OFF|ON|0|1}
```

```
DISPlay:P25:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

0 means that the graticule grid is hidden.

1 means that the graticule grid is showing.

## Examples

`DISPlay:P25:EDIagram:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the screen in the P25 Eye Diagram display.

## DISPlay:P25:EDIagram:Y[:SCALe]

Sets or queries the vertical scale (Hz) of the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Display commands

## Syntax

```
DISPlay:P25:EDIagram:Y[:SCALe]<NRf>
```

```
DISPlay:P25:EDIagram:Y[:SCALe]?
```

## Arguments

<value>::<NRf> specifies the vertical scale value (no units).

Range is 1 to 100

## Returns

<NR3> = a floating point value with an exponent, which is the vertical scale value in Hz.

## Examples

`DISP:P25:EDI:Y:SCAL 0.0005` sets the vertical scale to 500  $\mu$ Hz in the display.

`DISP:P25:EDI:Y:SCAL ?` might return `500.0000000000E-6`, indicating that the vertical scale is 500  $\mu$ Hz.

## **DISPlay:P25:EDIagram:Y[:SCALe]:AUTO (No Query Form)**

Sets the vertical scale automatically to fit the waveform to the screen in the P25 Eye Diagram display.

### **Conditions**

Measurement view: P25 Eye Diagram

### **Group**

Display commands

### **Syntax**

```
DISPlay:P25:EDIagram:Y[:SCALe]:AUTO
```

### **Arguments**

None.

### **Returns**

0 indicates the vertical scale is set to automatically fit the waveform to the screen in the P25 Eye Diagram display.

1 indicates the vertical scale is not set to automatically fit the waveform to the screen in the P25 Eye Diagram display.

### **Examples**

DISPLAY:P25:EDIAGRAM:Y[:SCALE]:AUTO sets the vertical scale automatically to fit the waveform to the screen.

## **DISPlay:P25:EDIagram:Y[:SCALe]:OFFSet**

Sets or queries the vertical offset (center point of the vertical axis) in the P25 Eye Diagram display.

### **Conditions**

Measurement view: P25 Eye Diagram

### **Group**

Display commands

### **Syntax**

```
DISPlay:P25:EDIagram:Y[:SCALe]:OFFSet
```

```
DISPlay:P25:EDIagram:Y[:SCALe]:OFFSet?
```

### **Arguments**

<value>::<NRf> specifies the minimum vertical value. Range is -50 to +50 (no units).

### **Returns**

Vertical offset value.

## Examples

`DISPLAY:P25:EDIAGRAM:Y[:SCALE]:OFFSET -0.5` sets the vertical offset to -0.5 in the P25 Eye Diagram display.

## DISPlay:P25:MEASview:DELeTe (No Query Form)

Deletes the specified P25 measurement display.

### Conditions

Measurement view: P25 measurements

### Group

Display commands

### Syntax

```
DISPlay:P25:MEASview:DELeTe { FDVT | EDI | PVT | CONS | SUMM }
```

### Related Commands

[DISPlay:P25:MEASview:NEW](#)

[DISPlay:P25:MEASview:SELeT](#)

### Arguments

FDVT: Frequency Deviation vs Time display

EDI: P25 Eye Diagram display

PVT: Power vs Time display

CONS: P25 Constellation display

SUMM: P25 Summary display

## Examples

`DISPLAY:P25:MEASVIEW:DELETE CONS` deletes the P25 Constellation display.

## DISPlay:P25:MEASview:NEW (No Query Form)

Displays a new P25 measurement view.

### Conditions

Measurement view: P25 measurements

### Group

Display commands

### Syntax

```
DISPlay:P25:MEASview:NEW { FDVT | EDI | PVT | CONS | SUMM }
```

## Related Commands

[DISPlay:P25:MEASview:SElect](#)

[DISPlay:P25:MEASview:DELeTe](#)

## Arguments

FDVT: Frequency Deviation vs Time display

EDI: P25 Eye Diagram display

PVT: Power vs Time display

CONS: P25 Constellation display

SUMM: P25 Summary display



**Note:** If you attempt to open a view that is currently displayed on the screen, the error -200, "Execution error; Measurement is already running" will be returned.

## Examples

`DISPLAY:P25:MEASVIEW:NEW PVT` creates a new view of the Power vs Time display.

## DISPlay:P25:MEASview:SElect

Selects a P25 measurement display. The query form returns the currently selected display.

## Conditions

Measurement view: P25 measurements

## Group

Display commands

## Syntax

```
DISPlay:P25:MEASview:SElect { FDVT | EDI | PVT | CONS | SUMM }
```

## Related Commands

[DISPlay:P25:MEASview:NEW](#)

[DISPlay:P25:MEASview:DELeTe](#)

## Arguments

FDVT: Frequency Deviation vs Time display

EDI: P25 Eye Diagram display

PVT: Power vs Time display

CONS: P25 Constellation display

SUMM: P25 Summary display



**Note:** If you attempt to select a view that is not displayed on the screen, the error -200, "Execution error; Measurement not running" will be returned.

## Returns

FDVT: Frequency Deviation vs Time display

EDI: P25 Eye Diagram display

PVT: Power vs Time display

CONS: P25 Constellation display

SUMM: P25 Summary display

## Examples

`DISPLAY:P25:MEASVIEW:SELECTCONS` selects the P25 Constellation display.

## DISPlay:P25:PVTime:BURSt:X[:SCALE]

Sets or queries the value of the scale (width) value, in seconds, for the P25 Power vs. Time display when using horizontal Full Burst view.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

`DISPlay:P25:PVTime:BURSt:X[:SCALE]`

`DISPlay:P25:PVTime:BURSt:X[:SCALE]?`

## Related Commands

[\*DISPlay:P25:PVTime:BURSt:X\[:SCALE\]:AUTO\*](#)

## Arguments

Floating point number that represents the value of the scale (width), in seconds, when using horizontal Full Burst view.

## Returns

Horizontal scale value.

## Examples

`DISPLAY:P25:PVTIME:BURST:X[:SCALE] 5.0E-6` sets the width of the display to 5.000  $\mu$ S for the horizontal Full Burst view.

## DISPlay:P25:PVTime:BURSt:X[:SCALE]:AUTO (No Query Form)

Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Full Burst view.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:BURSt:X[:SCALE]:AUTO
```

## Related Commands

[DISPlay:P25:PVTime:BURSt:X\[:SCALE\]](#)

## Examples

DISPLAY:P25:PVTIME:BURST:X[:SCALE]:AUTO automatically sets the starting time (position) and scale (width) values for the best display.

## DISPlay:P25:PVTime:BURSt:X[:SCALE]:OFFSet

Sets or queries the starting time (position) value, in seconds, for the P25 Power vs. Time display when using the horizontal Full Burst view.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:BURSt:X[:SCALE]:OFFSet
```

```
DISPlay:P25:PVTime:BURSt:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:P25:PVTime:BURSt:X\[:SCALE\]:AUTO](#)

## Arguments

Floating point number that represents the starting time (position) value, in seconds, when using horizontal Full Burst view.

## Returns

Horizontal offset value.

## Examples

DISPLAY:P25:PVTIME:BURST:X[:SCALE]:OFFSET 5.0E-6 sets the starting time (position) of the graph to 5.000  $\mu$ S for horizontal Full Burst view.

## DISPlay:P25:PVTime:BURSt:X[:SCALE]:RESet (No Query Form)

Resets the starting time (position) and scale (width) values for the horizontal Full Burst view on the P25 Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:BURSt:X[:SCALE]:RESet
```

## Related Commands

```
DISPlay:P25:PVTime:BURSt:X[:SCALE]:AUTO
```

## Examples

DISPLAY:P25:PVTIME:BURST:X[:SCALE]:RESET resets the starting time (position) and scale (width) values for the horizontal Full Burst view.

## DISPlay:P25:PVTime:FALL:X[:SCALE]

Sets or queries the scale (width) value, in seconds, for the P25 Power vs. Time display when using the horizontal Falling Edge view.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:FALL:X[:SCALE]<NRf>
```

```
DISPlay:P25:PVTime:FALL:X[:SCALE]?
```

## Related Commands

## Arguments

Floating point number that represents the scale (width), in seconds, when using the horizontal Falling Edge view.

## Returns

Horizontal scale value.

## Examples

DISPLAY:P25:PVTIME:FALL:X[:SCALE] 5.0E-6 sets the width of the graph to 5.000  $\mu$ S when using the horizontal Falling Edge view.

## **DISPlay:P25:PVTime:FALL:X[:SCALE]:AUTO (No Query Form)**

Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Falling Edge view.

### **Conditions**

Measurement view: P25 Power vs. Time

### **Group**

Display commands

### **Syntax**

DISPlay:P25:PVTime:FALL:X[:SCALE]:AUTO

### **Related Commands**

[DISPlay:P25:PVTime:FALL:X\[:SCALE\]:RESet](#)

### **Examples**

DISPLAY:P25:PVTIME:FALL:X[:SCALE]:AUTO automatically sets the starting time (position) and scale (width) values for the best display.

## **DISPlay:P25:PVTime:FALL:X[:SCALE]:OFFSet**

Sets or queries the value for the starting time (offset) of the graph, in seconds, for the P25 Power vs. Time display when using the horizontal Falling Edge view.

### **Conditions**

Measurement view: P25 Power vs. Time

### **Group**

Display commands

### **Syntax**

DISPlay:P25:PVTime:FALL:X[:SCALE]:OFFSet

### **Related Commands**

[DISPlay:P25:PVTime:RISE:X\[:SCALE\]](#)

### **Arguments**

Floating point number that represents the value of the starting time (offset) of the graph, in seconds, when using the horizontal Falling Edge view.

### **Returns**

Horizontal offset value.



## Examples

`DISPLAY:P25:PVTIME:FALL:X[:SCALE]:OFFSET 5.0` sets the starting time (position) of the graph to 5.000  $\mu$ S when using the horizontal Falling Edge view.

## DISPlay:P25:PVTime:FALL:X[:SCALE]:RESet (No Query Form)

Resets the starting time (position) and scale (width) values for the horizontal Falling Edge view on the P25 Power vs. Time display.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

`DISPlay:P25:PVTime:FALL:X[:SCALE]:RESet`

## Examples

`DISPLAY:P25:PVTIME:FALL:X[:SCALE]:RESET` resets the starting time (position) and scale (width) values for the horizontal Falling Edge view.

## DISPlay:P25:PVTime:MARKer:SHOW:STATe

Shows or hides the marker readout on the P25 Power vs. Time display.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

`DISPlay:P25:PVTime:MARKer:SHOW:STATe`

### Arguments

ON or 1 specifies to show the marker readout. OFF or 0 specifies to turn the marker readout off.

### Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`DISPLAY:P25:PVTIME:MARKER:SHOW:STATE?` might return #43204 (3204-byte data), which represents the vertical values (power) for the P25 Power vs. Time trace.

## **DISPlay:P25:PVTime:RISE:X[:SCALE]**

Sets or queries the scale (width) value, in seconds, for the P25 Power vs. Time display when using the horizontal Rising Edge view.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

```
DISPlay:P25:PVTime:RISE:X[:SCALE]
```

```
DISPlay:P25:PVTime:RISE:X[:SCALE]?
```

### Arguments

Floating point number that represents the scale (width) value, in seconds, for the horizontal Rising Edge view.

### Returns

Scale value.

## Examples

`DISPlay:P25:PVTIME:RISE:X[:SCALE] 5.0E-6` sets the width of the graph to 5.000  $\mu$ S for the horizontal Rising Edge view.

## **DISPlay:P25:PVTime:RISE:X[:SCALE]:AUTO (No Query Form)**

Sets the scale (width) value, in seconds, for the P25 Power vs. Time display to automatic.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

```
DISPlay:P25:PVTime:RISE:X[:SCALE]:AUTO
```

### Related Commands

[DISPlay:P25:PVTime:RISE:X\[:SCALE\]:RESet](#)

## Examples

`DISPLAY:P25:PVTIME:RISE:X[:SCALE]:AUTO` automatically sets the starting time (position) and scale (width) values for the best display.

## DISPlay:P25:PVTime:RISE:X[:SCALe]:OFFSet

Sets or queries the starting time value (offset), in seconds, for the P25 Power vs. Time display when using the horizontal Rising Edge view.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

`DISPlay:P25:PVTime:RISE:X[:SCALe]:OFFSet`

### Related Commands

[DISPlay:P25:PVTime:RISE:X\[:SCALe\]](#)

### Arguments

Floating point number that represents the value of the starting time (offset) of the graph, in seconds, for the horizontal Rising Edge view.

### Returns

Horizontal scale view

### Examples

`DISPLAY:P25:PVTIME:RISE:X[:SCALE]:OFFSET 5.0e-6` sets the starting time (position) of the graph to 5.000  $\mu$ S for the horizontal Rising Edge view.

## DISPlay:P25:PVTime:RISE:X[:SCALe]:RESet (No Query Form)

Resets the starting time (position) and scale (width) values for the horizontal Rising Edge view on the P25 Power vs. Time display.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Display commands

### Syntax

`DISPlay:P25:PVTime:RISE:X[:SCALe]:RESet`

## Related Commands

[DISPlay:P25:PVTTime:RISE:X\[:SCALE\]:OFFSet](#)

## Examples

`DISPLAY:P25:PVTIME:RISE:X[:SCALE]:RESET` resets the starting time (position) and scale (width) values for the horizontal Rising Edge view.

## DISPlay:P25:PVTTime:WINDow:SElect:PLOT

Sets or queries which view to use for the P25 Power vs. Time display. Select from Full Burst, Rising Edge or Falling Edge.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

`DISPlay:P25:PVTTime:WINDow:SElect:PLOT{BURSt | RISE | FALL}`

`DISPlay:P25:PVTTime:WINDow:SElect:PLOT?`

## Related Commands

## Arguments

- `BURSt` sets the view to Full Burst, which displays the entire packet, with vertical lines indicating the length of the burst.
- `RISE` sets the view to Rising Edge, which zooms the display into the interval around the burst rising edge, with vertical lines indicating the 10% to 90% Power-On Ramp time.
- `FALL` sets the view to Falling Edge, which zooms the display into the interval around the burst Falling Edge, with vertical lines indicating the 90% to 10% Power-Down Ramp time.

## Returns

`DISPlay:P25:PVTTime:WINDow:SElect:PLOT BURSt` sets the view to Full Burst.

## Examples

`DISPLAY:P25:PVTIME:WINDOW:SELECT:PLOT?` might return `RISE`, indicating that the view has been set to Rising Edge.

## DISPlay:P25:PVTTime:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries the graticule state in the Power vs Time display to on (showing) or off (hidden).

## Conditions

Measurement view: P25 Power vs Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:P25:PVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule in the Power vs. Time display.

ON or 1 shows the graticule in the Power vs. Time display.

## Returns

0 means the graticule in the Noise Temperature display is hidden (off).

1 means the graticule in the Noise Temperature display is showing (on).

## Examples

`DISPLAY:P25:PVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE ON` will turn on the graticule in the Noise Temperature display.

`DISPLAY:P25:PVTIME:WINDOW:TRACE:GRATICULE:GRID:STATE?` might return 1, indicating that the graticule in the Noise Temperature display is showing (on).

## DISPlay:P25:PVTime:Y[:SCALE]

Sets or queries the vertical scale value for the P25 Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:Y[:SCALE] <NRf>
```

```
DISPlay:P25:PVTime:Y[:SCALE]?
```

## Related Commands

```
DISPlay:P25:PVTime:Y[:SCALE]:PDIVision
```

## Arguments

<NRf> = a numeric value for the vertical scale in dB.

## Returns

<NR2> = a floating point number without an exponent, which is the vertical scale value.

## Examples

DISP:P25:PVT:Y:SCAL 10.5 sets the vertical scale value to 10.50 dB.

DISP:P25:PVT:Y:SCAL ? might return 10, indicating that the vertical scale is 10.00 dB.

## DISPlay:P25:PVTime:Y[:SCALe]:AUTO (No Query Form)

Automatically selects the vertical scale and position values for the P25 Power vs Time display.

## Conditions

Measurement view: P25 Power vs Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:Y[:SCALe]:AUTO
```

## Related Commands

[DISPlay:P25:PVTime:Y\[:SCALe\]:AUTO](#)

## Examples

DISP:P25:PVT:Y:SCAL:AUTO specifies to reset the vertical scale and position (offset) values for the P25 Power vs. Time display.

## DISPlay:P25:PVTime:Y[:SCALe]:OFFSet

Sets or queries the value of the vertical offset (top edge of the vertical axis) for the P25 Power vs. Time display.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

```
DISPlay:P25:PVTime:Y[:SCALe]:OFFSet<NRf>
```

```
DISPlay:P25:PVTime:Y[:SCALe]:OFFSet?
```

## Related Commands

[DISPlay:P25:PVTime:Y\[:SCALE\]](#)

## Arguments

Floating point number between limits that specifies the value of the vertical offset. Positive inputs move the plot down and negative inputs move the plot up (as viewed by the user).

## Returns

The value of the vertical offset for the P25 Power vs. Time display as a floating point number.

## Examples

DISPLAY:P25:PVTIME:Y[:SCALE]:OFFSET -10 sets the vertical offset to -10 dBm.

## DISPlay:P25:PVTime:Y[:SCALE]:PDIVision

Sets or queries the value of the vertical scale for the P25 Power vs. Time display, in dB/division. This is only a visual control for panning the graph.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Display commands

## Syntax

DISPlay:P25:PVTime:Y[:SCALE]:PDIVision<NRf>

DISPlay:P25:PVTime:Y[:SCALE]:PDIVision?

## Related Commands

[DISPlay:P25:PVTime:Y\[:SCALE\]](#)

## Arguments

Floating point value that represents the value of the vertical scale, in dB/division, for the P25 Power vs. Time display.

## Examples

DISPLAY:P25:PVTIME:Y[:SCALE]:PDIVISION 20 sets the vertical scale to 20.0 dB/division.

## DISPlay:P25:RADix

Set or queries the symbol radix for the P25 Symbol Table display.

## Conditions

Measurement view: P25 Symbol Table

## Group

Display commands

## Syntax

```
DISPlay:P25:RADix { QUAT | BIN | MODS }
```

## Arguments

QUATernary: sets the symbols base to quaternary.

BINary: sets the symbols base to binary.

MODsymbols: sets the symbols to modular.

## Returns

QUAT: means that the symbols base is set to Quaternary.

BINary: means that the symbols base is set to Binary.

MODsymbols: means that the symbols base is set to ModSymbols (modular).

## Examples

```
DISPLAY:P25:RADIX
```

## DISPlay:PERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE

Sets or queries the graticule grid view state for the Phase error versus Time view.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Display commands

## Syntax

```
DISPlay:PERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE { OFF | ON | 0 | 1 }  
DISPlay:PERRor[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.



## Returns

OFF or 0

ON or 1

## Examples

`DISPlay:PERRor:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON` shows the graticule grid on the Phase error versus Time Measview 2.

## DISPlay:PERRor[:MEASview<y>]:Y[:SCALe]

Sets or queries the vertical range of the Phase error versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Display commands

## Syntax

`DISPlay:PERRor[:MEASview<y>]:Y[:SCALe] <value>`

`DISPlay:PERRor[:MEASview<y>]:Y[:SCALe]?`

## Related Commands

[\*DISPlay:PERRor\[:MEASview1\]:Y\[:SCALe\]:OFFSet\*](#)

## Arguments

<value>::<NRf> specifies the vertical range.

Range: 1 to 100%.

## Returns

<NRf>

## Examples

`DISPlay:PERRor:MEASview1:Y:SCALe` sets the vertical range to 50% in the Phase error versus Time display Measurement view 1.

## DISPlay:PERRor[:MEASview<y>]:Y[:SCALe]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the Phase error versus Time display.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### Group

Display commands

### Syntax

```
DISPlay:PERRor[:MEASview<y>]:Y[:SCALE]:AUTO
```

### Arguments

None

### Examples

DISPlay:PERRor:MEASview2:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen of Phase error versus Time measurement view 2.

## DISPlay:PERRor[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the minimum vertical value (bottom edge) of the Phase error versus Time graph.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### Group

Display commands

### Syntax

```
DISPlay:PERRor[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:PERRor[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

### Related Commands

[DISPlay:PERRor\[:MEASview<y>\]:Y\[:SCALE\]](#)

### Arguments

<value>::<NRf> specifies the minimum vertical value.

Range: -360 to +360°.

## Returns

<NRf>

## Examples

DISPlay:PErRor:MEASview2:Y:SCALE:OFFSet? -9.5 sets the minimum vertical value to -9.5% in the Phase error versus Time graph Measurement view 2.

## DISPlay:PHVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

```
DISPlay:PHVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PHVTime[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

DISPLAY:PHVTIME:MEASview1:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the Frequency versus Time view.

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALE]

Sets or queries the horizontal scale (full-scale time) of the Phase versus Time graph. Programming a specified scale sets DISPlay:PHVTime[:MEASview<y>]:X[:SCALE] AUTO:STATe OFF.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALE] <value>
```

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALE]?
```

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe](#), [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <NRf> | MAXimum | MINimum } specifies the horizontal scale in full-scale time. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALE\]:MAXimum?](#) and [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALE\]:OFFSet:MINimum?](#) queries to get the upper and lower limit values of the setting range.

## Examples

```
DISPLAY:PHVTIME:MEASview1:X:SCALE 1.5ms sets the horizontal scale to 1.5 ms.
```

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale automatically to fit the waveform to the screen in the Phase versus Time view. Executing this command sets DISPlay:PHVTime[:MEASview<y>]:X[:SCALE]:AUTO:STATe ON.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALE]:AUTO
```

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALE\]:AUTO:STATe](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

```
DISPLAY:PHVTIME:MEASview1:X:SCALE:AUTO sets the horizontal scale automatically to fit the waveform to the screen.
```

## DISPlay:PHVTime[:MEASview?]:Y[:SCALe]:AXIS:ENABLE:REFerence {set and query}

Sets or queries the state of Y reference checkbox.

### Conditions

Measurement views: Phase VS Time.

Applicable only for MSO/LPD/SX.

### Group

Display commands

### Syntax

```
DISPlay:PHVTime[:MEASview?]:Y[:SCALe]:AXIS:ENABLE:REFerence {0|1|TRUE|FALSE}
```

```
DISPlay:PHVTime[:MEASview?]:Y[:SCALe]:AXIS:ENABLE:REFerence? (Query)
```

### Arguments

0 is used to change the 'Y Reference' checkbox state from checked to unchecked.

1 is used to change the 'Y Reference' checkbox state from unchecked to checked.

### Example

DISPlay:PHVTime:MEASview1:Y:SCALe:AXIS:ENABLE:REFerence 1 changes the state of 'Y Reference' checkbox to checked.

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe

Determines whether to set the horizontal scale automatically or manually.

### Conditions

Measurement views: Phase versus Time

### Group

Display commands

### Syntax

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:AUTO:STATe?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the horizontal scale is set manually. To set it, use the [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]](#) and [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet](#) commands.

ON or 1 specifies that the horizontal scale is set automatically.

## Examples

`DISPLAY:PHVTIME:MEASview1:X:SCALE:AUTO:STATE ON` specifies that the horizontal scale is set automatically.

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MAXimum? (Query Only)

Queries the upper limit of the horizontal scale setting range.

### Conditions

Measurement views: Phase versus Time

### Group

Display commands

### Syntax

`DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MAXimum?`

### Related Commands

[\*DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]\*](#)

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

<NRf> The upper limit of the horizontal scale setting range.

### Examples

`DISPLAY:PHVTIME:MEASview1:X:SCALE:MAXIMUM?` might return 18.135E-3, indicating that the upper limit of the horizontal scale setting range is 18.135 ms.

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MINimum? (Query Only)

Queries the lower limit of the horizontal scale setting range.

### Conditions

Measurement views: Phase versus Time

### Group

Display commands

### Syntax

`DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:MINimum?`

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<NRf> The lower limit of the horizontal scale setting range.

## Examples

DISPLAY:PHVTIME:MEASview1:X:SCALE:MINIMUM? might return 10.0E-9, indicating that the lower limit of the horizontal scale setting range is 10.0 ns.

## DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the Phase versus Time graph. Programming a specified offset sets DISPlay:PHVTime:MEASview<y>:X[SCALe]:AUTO:STATe OFF.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet

DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet?

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:AUTO:STATe](#), [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]](#),

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::={ <NRf> | MAXimum | MINimum } specifies the horizontal offset. MAXimum and MINimum represent the upper and lower limits of the setting range, respectively.

Use the [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet:MAXimum?](#) and [DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet:MINimum?](#) queries to get the upper and lower limit values of the setting range.

## Examples

DISPLAY:PHVTIME:MEASview1:X:SCALE:OFFSET 800ns sets the minimum horizontal value to 800 ns in the Phase versus Time graph.

## **DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum? (Query Only)**

Queries the upper limit of the horizontal offset setting range.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MAXimum?
```

### **Related Commands**

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet](#)

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Returns**

<NRf> The upper limit of the horizontal offset setting range.

### **Examples**

DISPLAY:PHVTIME:MEASview1:X:SCALE:OFFSET:MAXIMUM? might return -1.812E-3, indicating that the upper limit of the horizontal offset setting range is -1.812 ms.

## **DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum? (Query Only)**

Queries the lower limit of the horizontal offset setting range.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:PHVTime[:MEASview<y>]:X[:SCALe]:OFFSet:MINimum?
```

### **Related Commands**

[DISPlay:PHVTime\[:MEASview<y>\]:X\[:SCALe\]:OFFSet](#)



## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<NRf> The lower limit of the horizontal offset setting range.

## Examples

DISPLAY:PHVTIME:MEASview1:X:SCALE:OFFSET:MINIMUM? might return -16.28E-3, indicating that the lower limit of the horizontal offset setting range is -16.28 ms.

## DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]

Sets or queries the vertical range of the Phase versus Time graph.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE] <value>

DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]?

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:Y\[:SCALE\]:OFFSet](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the vertical range. Range: 1 to 1T °.

## Examples

DISPLAY:PHVTIME:MEASview1:Y:SCALE 180 sets the vertical range to 180 ° in the Phase versus Time graph.

## DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the Phase versus Time view.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AUTO

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

DISPLAY:PHVTIME:MEASview1:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS

Sets or queries the vertical axis representation.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS { MODulo $\pi$  | CONTinuous }

DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

MODulo $\pi$  (modulo  $\pi$ ) shows the phase constrained within  $\pm 180^\circ$  along the vertical axis.

CONTinuous shows the phase as continuous quantity along the vertical axis.

## Examples

DISPLAY:PHVTIME:MEASview1:Y:SCALE:AXIS MODulo $\pi$  selects modulo  $\pi$  representation for the vertical axis.

## DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS:REFerence

Sets or queries which time point in the analysis period to use as the zero-phase-value reference.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

```
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS:REfERENCE <value>
```

```
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:AXIS:REfERENCE?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the phase reference time.

## Examples

DISPLAY:PHVTIME:MEASview1:Y:SCALE:AXIS:REFERENCE 1.5us sets the phase reference time to 1.5  $\mu$ s.

## DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the center of the vertical axis) in the Phase versus Time graph.

## Conditions

Measurement views: Phase versus Time

## Group

Display commands

## Syntax

```
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:PHVTime\[:MEASview<y>\]:Y\[:SCALE\]](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the vertical offset. Range: -0.5T to +0.5T°.

## Examples

DISPLAY:PHVTIME:MEASview1:Y:SCALE:OFFSET -158.5 sets the vertical offset to -158.5 ° in the Phase versus Time graph.

## **DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:RESCale (No Query Form)**

Rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:PHVTime[:MEASview<y>]:Y[:SCALE]:RESCale
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Examples**

DISPLAY:PHVTIME:MEASview1:Y:SCALE:RESCALE rescales the vertical axis automatically to fit the Phase versus Time waveform to the screen.

## **DISPlay:PNOise:LEGend:STATe**

Determines whether to show or hide the trace legend on the display. The legend indicates the trace detection and function on the screen for each displayed trace.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:PNOise:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PNOise:LEGend:STATe?
```

### **Arguments**

OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

### **Examples**

DISPLAY:PNOISE:LEGEND:STATE ON shows the trace legend on the screen.

## DISPlay:PNOise:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the phase noise view.

### Conditions

Measurement views: Phase noise

### Group

Display commands

### Syntax

```
DISPlay:PNOise:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PNOise:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

### Examples

DISPLAY:PNOISE:MARKER:SHOW:STATE ON shows the readout for the selected marker in the graph.

## DISPlay:PNOise:RESet:SCALE (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the phase noise view.

Vertical offset = -50 dBc/Hz, Vertical scale = 100 dB, Horizontal start = 10 Hz, and Horizontal stop = 1 GHz

### Conditions

Measurement views: Phase noise

### Group

Display commands

### Syntax

```
DISPlay:PNOise:RESet:SCALE
```

### Arguments

None

### Examples

DISPLAY:PNOISE:RESET:SCALE resets the horizontal and vertical scale to the default values.

## DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PNOise:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

```
DISPLAY:PNOISE:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the screen.
```

## DISPlay:PNOise:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the phase noise view.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:X[:SCALE]:AUTO
```

## Arguments

None

## Examples

```
DISPLAY:PNOISE:X:SCALE:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.
```

## DISPlay:PNOise:X[:SCALE]:START

Sets or queries the start frequency (left edge) of the phase noise graph.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:X[:SCALE]:START <value>
```

```
DISPlay:PNOise:X[:SCALE]:START?
```

## Arguments

<value>::<NRf> specifies the start frequency.

Range: 10 mHz to 100 MHz.

Note that (start frequency) = 104 × (stop frequency).

## Examples

```
DISPLAY:PNOISE:X:SCALE:START 10Hz sets the start frequency to 10 Hz in the phase noise graph.
```

## DISPlay:PNOise:X[:SCALE]:STOP

Sets or queries the stop frequency (right edge) of the phase noise graph.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:X[:SCALE]:STOP <value>
```

```
DISPlay:PNOise:X[:SCALE]:STOP?
```

## Arguments

<value>::<NRf> specifies the stop frequency.

Range: 100 Hz to 1 THz.

Note that (start frequency) = 104 × (stop frequency).

## Examples

```
DISPLAY:PNOISE:X:SCALE:STOP 2GHz sets the stop frequency to 2 GHz in the phase noise graph.
```

## DISPlay:PNOise:Y[:SCALE]

Sets or queries the vertical range of the phase noise graph.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:Y[:SCALE] <value>
```

```
DISPlay:PNOise:Y[:SCALE]?
```

## Arguments

<value>::<Nrf> specifies the vertical range. Range: 0.1 to 200 dB.

## Examples

DISPLAY:PNOISE:Y:SCALE 100 sets the vertical range to 100 dB for the phase noise graph.

## DISPlay:PNOise:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the phase noise view.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:PNOISE:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:PNOise:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the phase noise graph.

## Conditions

Measurement views: Phase noise



## Group

Display commands

## Syntax

```
DISPlay:PNOise:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:PNOise:Y[:SCALE]:OFFSet?
```

## Arguments

<value>::<Nrf> specifies the vertical offset. Range: -200 to +20 dBc/Hz.

## Examples

DISPLAY:PNOISE:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBc/Hz for the phase noise graph.

## DISPlay:PNOise:Y[:SCALE]:PDIVision

Sets or queries the vertical scale (per division) of the phase noise graph.

## Conditions

Measurement views: Phase noise

## Group

Display commands

## Syntax

```
DISPlay:PNOise:Y[:SCALE]:PDIVision <value>
```

```
DISPlay:PNOise:Y[:SCALE]:PDIVision?
```

## Arguments

<value>::<Nrf> specifies the vertical scale (per division).

Range: 0.01 to 20 dB/div.

## Examples

DISPLAY:PNOISE:Y:SCALE:PDIVISION 5 sets the vertical scale to 5 dB/div.

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:CURRent:BIN

Sets or queries the current bin number where the indicator is positioned from the histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:CURRent:BIN <value>
```

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:CURRent:BIN?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NR1> sets the current bin number.

## Examples

```
DISPLAY:PULSE:CUMULATIVE:HISTOGRAM:CURRENT:BIN 50 sets the current bin number to 50.
```

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:AUTO (No Query Form)

Sets the scale auto value in the histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:AUTO <value>
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NR1>

## Examples

```
DISPLAY:PULSE:CUMULATIVE:HISTOGRAM:MEASview<y>:SCALE:AUTO1 sets the auto scale value to 1.
```

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:RESEt (No Query Form)

Resets the current scale in the histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>:]SCALE:RESEt <value>
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NR1>

## Examples

```
DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:RESEt sets the scale value to 1.
```

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:AUTO (No Query Form)

Sets the X auto value in the histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:AUTO <value>
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

## Returns

<NR1>

## Examples

```
DISPLAY:PULSE:CUMULATIVE:HISTOGRAM:MEASview<y>:SCALE:X:AUTO 1 sets the X auto value to 1.
```

## **DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:NUMBER**

Sets or queries the current X scale number from the histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:NUMBER <value>
```

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:NUMBER?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

### **Returns**

<NR1> returns the X scale value.

### **Examples**

```
DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:X:NUMBER 50 sets the X scale value to 50.
```

## **DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:OFFSET**

Sets or queries the current X offset number from the histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:OFFSET <value>
```

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:OFFSET?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

**Returns**

<NR1> returns the X offset value.

**Examples**

DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:X:OFFSET 2 sets the X scale offset value to 2.

**DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:RScale (No Query Form)**

Sets the current X reset scale in the histogram display.

**Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

**Group**

Display commands

**Syntax**

DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:X:RScale <value>

**Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid.

**Returns**

<NR1>

**Examples**

DISPLAY:PULSE:CUMULATIVE:HISTOGRAM:MEASview<y>:SCALE:X:RSCALE 1 sets the X reset scale value to 1.

**DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:AUTO (No Query Form)**

Sets the current Y auto value in the histogram display.

**Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

**Group**

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:AUTO <value>
```

## Arguments

<value> sets the Y auto value.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NR1>

## Examples

```
DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:Y:AUTO 1 sets the Y auto value to 1.
```

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:FULL

Sets or queries the current Y scale full number from the histogram display.

## Conditions

Measurement views:

Cumulative Histogram with the specified Measurement View Number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:FULL <value>
```

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:FULL?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NR1> sets the Y scale full value

## Examples

```
DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:Y:FULL? 10 sets the Y scale value to 10.
```

## DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:OFFSet

Sets the current Y offset numbering the histogram display.

**Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

**Group**

Display commands

**Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:OFFSet<value>
```

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:OFFSet?
```

**Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

**Returns**

<NR1> returns the Y scale offset value.

**Examples**

DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:Y:OFFSET 2 sets the Y scale offset value to 2.

**DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:RSCale (No Query Form)**

Sets the current Y reset scale in the histogram display.

**Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

**Group**

Display commands

**Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:RSCale <value>
```

**Arguments**

<value> sets the Y reset scale value.

**Returns**

<NR1> sets the Y reset scale value.

**Examples**

DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:Y:RSCALE 1 sets the Y reset scale value to 1.

## **DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:STOP? (Query Only)**

Queries the minimum vertical value (bottom edge) in the cumulative histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:SCALE:Y:STOP?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf> is the minimum vertical value (bottom edge).

### **Examples**

DISPlay:PULSe:CUMulative:HISTogram:MEASview<y>:SCALE:Y:STOP? might return 0, indicating that the minimum vertical value is 0 in the pulse cumulative histogram display.

## **DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Sets or queries the current graticule selection from the histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:HISTogram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe  
{ 0 | 1 }
```

### **Arguments**

0 sets the display without graticule (box is unchecked)

1 sets the display with graticule (checkbox is checked)



The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

0 or 1, <NRf>

## Examples

DISPLAY:PULSE:CUMULATIVE:HISTOGRAM:MEASview<y>:WINDOW:TRACE:GRATICULE:GRID:STATE1 sets the display with graticule option checked.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFReqerror

Sets or queries the Max Frequency error measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFReqerror <value>

DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFReqerror

DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFReqerror?

## Arguments

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:MFReqerror 1 enables the Max Frequency Error measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:ATX

Sets or queries the Peak Power measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:ATX <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:ATX?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:ATX 1 enables the Average Transmitted Power measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:AVERage

Sets or queries the Average ON Power measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:AVERage <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:AVERage?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:AVERage 1` enables the Average ON Power measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DRODb**

Sets or queries the Droop dB measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DRODb <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DRODb?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:DRODb 1` enables the Droop dB measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DROOp**

Sets or queries the Droop measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DROOp <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DROOp?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:DROop 1 enables the Droop measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTPct

Sets or queries the Duty Factor (%) measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTPct <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTPct?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:DUTPct 1 enables the Duty Factor(%) measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTRatio

Sets or queries the Duty Factor (Ratio) measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTRatio <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:DUTRatio?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:DUTRatio 1 enables the Duty Factor (Ratio) measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FABS

Sets or queries the Absolute Frequency measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FABS <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FABS?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:FABS 1` enables the Absolute Frequency measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FALL**

Sets or queries the Fall Time measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FALL <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FALL?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:FALL 1` enables the Fall Time measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FDELta**

Sets or queries the Delta Frequency measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FDELta <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FDELta?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRF>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:FDELta 1 enables the Delta Frequency measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FRDeviation

Sets or queries the Frequency Deviation measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FRDeviation <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:FRDeviation?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRF>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:FRDeviation 1 enables the Frequency Deviation measurement.

## **DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRAMplitude**

Sets or queries the Response Amplitude measurement selection from the Cumulative Statistics display.

### **Conditions**

Measurement views: Pulse Cumulative Statistics

### **Group**

Display commands

### **Syntax**

```
DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRAMplitude <0 | 1>
```

```
DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRAMplitude?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### **Returns**

0 or 1, <NRf>

### **Examples**

DISPLAY:PULSe:CUMulative:STATistics:MEASview1:IRAMplitude 1 enables the Response Amplitude measurement.

## **DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRTime**

Sets or queries the Impulse Response Time measurement selection from the Cumulative Statistics display.

### **Conditions**

Measurement views: Pulse Cumulative Statistics

### **Group**

Display commands

### **Syntax**

```
DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRTime <0 | 1>
```

```
DISPLAY:PULSe:CUMulative:STATistics[:MEASview<y>]:IRTime?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.



0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPLAY:PULSE:CUMulative:STATistics:MEASview1:IRTime 1` enables the Impulse Response Time measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MPHerror

Sets or queries the Max phase error measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MPHerror <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MPHerror?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 or OFF unchecks the checkbox

1 or ON checks the checkbox

## Returns

<Boolean> , 0 or 1

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:MPHerror 1` enables the Max phase error measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVEdB

Sets or queries the Overshoot dB measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVEDb <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVEDb?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

```
DISPlay:PULSe:CUMulative:STATistics:MEASview1:OVEDb 1 enables the Overshoot dB measurement.
```

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVERshoot

Sets or queries the Overshoot measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVERshoot <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:OVERshoot?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:OVERshoot 1` enables the Overshoot measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PHDeviation

Sets or queries the Phase Deviation measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PHDeviation <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PHDeviation?`

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:PHDeviation 1` enables the Phase Deviation measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFD

Sets or queries the Pulse-Pulse Frequency Difference measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFD <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFD?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:PPFD 1 enables the Pulse-Pulse Frequency Difference measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFrequency

Sets or queries the Pulse-pulse carrier frequency measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFrequency <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPFrequency?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:PPFrequency 1 enables the Pulse-pulse carrier frequency phase measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPOWER

Sets or queries the Peak Power measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPOWer <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPOWer?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:PPOWer 1 enables the Peak Power measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPD

Sets or queries the Pulse-Pulse Phase Difference measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPD <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPD?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:PPPD 1` enables the Pulse-Pulse Phase Difference measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPHase**

Sets or queries the Pulse-pulse carrier phase measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPHase <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:PPPHase?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:PPPHase 1` enables the Pulse-pulse carrier phase measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RINTerval**

Sets or queries the Repetition Interval measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RINTerval <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RINTerval?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:RINTerval 1 enables the Repetition Interval measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPDb

Sets or queries the Ripple dB measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPDb <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPDb?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:RIPDb 1 enables the Ripple dB measurement.

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPPlE**

Sets or queries the Ripple measurement selection from the Cumulative Statistics display.

### **Conditions**

Measurement views: Pulse Cumulative Statistics

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPPlE <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RIPPlE?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### **Returns**

0 or 1, <NRf>

### **Examples**

```
DISPlay:PULSe:CUMulative:STATistics:MEASview1:RIPPlE 1 enables the Ripple measurement.
```

## **DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RISE**

Sets or queries the Rise Time measurement selection from the Cumulative Statistics display.

### **Conditions**

Measurement views: Pulse Cumulative Statistics

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RISE <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RISE?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.



0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:RISE 1` enables the Rise Time measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSFreqerror

Sets or queries the RMS Frequency error measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSFreqerror <0 | 1>`

`DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSFreqerror?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:RMSFreqerror 1` enables the RMS Frequency error measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFRreqerror

Sets or queries the Max Frequency error measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFRerror <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:MFRerror?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox.

1 checks the checkbox.

## Returns

<0 | 1>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:MFRerror 1 enables the Max Frequency Error measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSPherror

Sets or queries the Max Frequency error measurement selection from the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSPherror <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RMSPherror?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:RMSPherror 1` enables the Max Frequency Error measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RRATe

Sets or queries the Repetition Rate measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RRATe <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:RRATe?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### Returns

0 or 1, <NRf>

## Examples

`DISPlay:PULSe:CUMulative:STATistics:MEASview1:RRATe 1` enables the Repetition Rate measurement.

## DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:WIDTh

Sets or queries the Pulse Width measurement selection from the Cumulative Statistics display.

### Conditions

Measurement views: Pulse Cumulative Statistics

### Group

Display commands

### Syntax

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:WIDTh <0 | 1>
```

```
DISPlay:PULSe:CUMulative:STATistics[:MEASview<y>]:WIDTh?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

## Returns

0 or 1, <NRf>

## Examples

DISPlay:PULSe:CUMulative:STATistics:MEASview1:WIDTh 1 enables the Pulse Width measurement.

## DISPlay:PULSe:MEASview<y>:DELeTe (No Query Form)

Deletes the measurement view in the pulsed RF measurements.

## Conditions

Measurement views: Pulsed RF measurements

## Group

Display commands

## Syntax

```
DISPlay:PULSe:MEASview<y>:DELeTe { RESult | TRACe | STATistics }
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

RESult deletes the pulse table view.

TRACe deletes the pulse trace view.

STATistics deletes the pulse statistics view.

If you attempt to delete a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Examples

DISPlay:PULSe:MEASview1:DELeTe deletes the pulse trace view.

## DISPlay:PULSe:MEASview<y>[:SOURce<x>]:NEW (No Query Form)

Selects a measurement view in the pulsed RF measurements on the screen. The query command returns the currently selected view.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

Measurement views: Pulsed RF with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:MEASview<y>[:SOURce<x>]:NEW { RESult | TRACe | STATistics }
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<x> = 1 to 8, representing Range of Source.

RESult opens the pulse table view.

TRACe opens the pulse trace view.

STATistics opens the pulse statistics view.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Examples

```
DISPlay:PULSe:MEASview2:SOURce1:NEW STATistics creates pulse statistics measurement with source 1.
```

## DISPlay:PULSe:MEASview<y>:SElect

Selects a measurement view in the pulsed RF measurements on the screen. The query command returns the currently selected view.

## Conditions

Measurement views: Pulsed RF with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:MEASview<y>:SElect { RESult | TRACe | STATistics }
```

```
DISPlay:PULSe:MEASview<y>:SElect?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

RESult selects the pulse table view.

TRACe selects the pulse trace view.

STATistics selects the pulse statistics view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement not running") will be returned.

## Returns

<Nrf>

## Examples

DISPlay:PULSe:MEASview2:SElect TRACe selects the pulse trace Measurement view 2.

## DISPlay:PULSe:OGRAM[:MEASview<y>]

Sets the chosen display from the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

DISPlay:PULSe:OGRAM[:MEASview<y>]<arg1>

DISPlay:PULSe:OGRAM[:MEASview<y>]

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<arg1> is FVT or TVT or BOTH

Where

TVT = Time Vs Time

FVT = Frequency Vs Time

BOTH = Both

## Returns

<String> FVT or TVT or BOTH

## Examples

DISPlay:PULSe:OGRAM:MEASview1 BOTH sets the display type to both.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:FRAME:INFORmation

Sets or queries the Frame Info parameter is enabled or not from the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:FRAME:INFORmation <0 | 1>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:FRAME:INFORmation?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 unchecks the checkbox

1 checks the checkbox

### Returns

0 or 1, <NRf>

### Examples

DISPlay:PULSe:OGRAM:MEASview1:FRAME:INFORmation 1 enables the Frame Info parameter.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:FRAME

Sets or queries the Marker frame number from the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:FRAME <value>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:FRAME?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the marker frame number.

## Returns

<NR1>

## Examples

DISPlay:PULSe:OGRAM:MEASview1:MARKer:FRAME 5 sets the marker frame number to 5.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:SHOW:STATE

Sets or queries the Show Marker readout in graph option from the Prefs tab of the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:SHOW:STATE {OFF | ON | 0 | 1}

DISPlay:PULSe:OGRAM[:MEASview<y>]:MARKer:SHOW:STATE?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

0 or OFF unchecks the checkbox.

1 or ON checks the checkbox.

## Returns

{OFF | ON | 0 | 1}

## Examples

DISPlay:PULSe:OGRAM:MEASview1:MARKer:SHOW:STATE 1 enables the Show Marker readout in graph parameter.

## DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:START

Sets or queries the current X scale frequency start value from the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands



## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:START <value>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:START?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the X scale frequency start value.

## Returns

<NR1>

## Examples

DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:FVTime:FREQuency:START 43 sets X scale frequency start value to 43.

## DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:STOP

Sets or queries the current X scale frequency stop value from the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:STOP <value>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:FREQuency:STOP?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the X scale frequency stop value.

## Returns

<NR1>

## Examples

DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:FVTime:FREQuency:STOP 155 sets X scale frequency stop value to 155.

## **DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:RSCale (No Query Form)**

Restores the horizontal axis of start frequency and stop frequency to default value in the Frequency vs Time plot of Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:FVTime:RSCale
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Examples**

DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:FVTime:RSCale restores the horizontal axis of start frequency and stop frequency to default value.

## **DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:RSCale (No Query Form)**

Restores the horizontal axis of position and scale to default value in the Time vs Time plot of Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:RSCale
```

### **Arguments**

None

### **Examples**

DISPlay:PULSe:OGRAM:MEASview1[:SCALE]:X:RSCALE restores the horizontal axis of position and scale to default value.

## **DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime (No Query Form)**

Sets the scale value in the Time vs Time of Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime <value>
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the scale value in Time vs Time.

## Examples

```
DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:TVTime 1 sets the scale value to 1.
```

## DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime:OFFSet (No Query Form)

Sets the scale value in the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime:OFFSet <value>
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the X scale offset value.

## Examples

```
DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:TVTime:OFFSet 1 sets the X scale offset value to 1.
```

## DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime:RSCale (No Query Form)

Restores the horizontal axis of position and scale to default value in the Time vs Time plot of Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:X:TVTime:RSCale
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

None

## Examples

DISPlay:PULSe:OGRAM:MEASview1:SCALE:X:TVTime:RSCale restores the horizontal axis of position and scale to default value.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:FVTime:DETection

Sets or queries the type of detection for the specified trace.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:FVTime:DETection {AVERage | POSitive | NEGative | POSNegative | SAMPlE}
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:FVTime:DETection?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

AVERage displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPLE displays the first sample value received for each pixel.

### Returns

{AVERage | POSitive | NEGative | POSNegative | SAMPlE}

### Examples

DISPlay:PULSe:OGRAM:MEASview1:FVTime:DETEctionPOSitive enables positive detection.

## DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:RSCale

Restores the default value of both the axis inTime vs Time and Frequency vs Time plot of Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:RSCale

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Examples

DISPlay:PULSe:OGRAM:MEASview1:SCALE:RSCale restores default value in PulseOgram displays.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATE

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

DISPlay:PULSe:OGRAM[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATE { OFF | ON | 0 | 1 }

DISPlay:PULSe:OGRAM[:MEASview<y>]:WINDow:TRACe:GRATICule:GRID:STATE?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

{ OFF | ON | 0 | 1 }

## Examples

DISPlay:PULSe:OGRAM:MEASview1:WINDow:TRACe:GRATicule:GRID:STATe ON shows the graticule grid on the screen.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:TRACE:POINTs

Sets or queries the trace points in the Pulseo-gram measurement.

## Conditions

Measurement views: Pulse-Ogram

## Group

Display commands

## Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:TRACE:POINTs {ONEK | TENK | HUNDredk | NDECimate | NEVerdecimate}
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:TRACE:POINTs?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

ONEK sets the trace points to 1k.

TENK sets the trace points to 10k.

HUNDredk sets the trace points to 100k.

NDECimate or NEVerdecimate sets the trace points to Never decimate.

## Returns

{ONEK | TENK | HUNDredk | NDECimate | NEVerdecimate}

## Examples

DISPlay:PULSe:OGRAM:MEASview1:TRACE:POINTs ONEK selects the trace points of 1000 in the PulseOgram measurement.

**DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:Y:FRAME:FULL**

Sets or queries the frame scale value in the Pulse-Ogram display in fast frame mode acquisition.

**Conditions**

Measurement views: Pulse-Ogram

**Group**

Display commands

**Syntax**

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:Y:FRAME:FULL <value>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>][:SCALE]:Y:FRAME:FULL?
```

**Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the frame scale value.

**Returns**

<NR1> sets the frame scale value.

**Examples**

```
DISPlay:PULSe:OGRAM:MEASview1:SCALE:Y:FRAME:FULL 4 sets the frame scale value to 4.
```

**DISPlay:PULSe:OGRAM:SCALE:Y:FRAME:OFFSet**

Sets or queries the frame offset value in the Pulse-Ogram display in fast frame mode acquisition.

**Conditions**

Measurement views: Pulse-Ogram

**Group**

Display commands

**Syntax**

```
DISPlay:PULSe:OGRAM:SCALE:Y:FRAME:OFFSet <value>
```

```
DISPlay:PULSe:OGRAM:SCALE:Y:FRAME:OFFSetDISPlay:PULSe:OGRAM: SCALE:Y:FRAME:OFFSet?
```

**Arguments**

<value> :: = <NR1> sets the frame offset value.

## Examples

DISPLAY:PULSE:OGRAM:SCALE:Y:FRAME:OFFSET 2 sets the frame offset value to 2.

## DISPlay:PULSe:OGRAM[:MEASview<y>]:TVTime:SMOoth:POINTs

Sets or queries the value of Smooth points from the Traces tab in Time vs Time display of the Pulse-Ogram.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:TVTime:SMOoth:POINTs <value>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]:TVTime:SMOoth:POINTs?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> :: = <NR1> sets the smooth points value.

### Returns

<NR1> sets the smooth points value.

## Examples

DISPlay:PULSe:OGRAM:MEASview1:TVTime:SMOoth:POINTs 5 sets the smooth points value to 5.

## DISPlay:PULSe:OGRAM[:MEASview<y>]

Sets the chosen display from the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Display commands

### Syntax

```
DISPlay:PULSe:OGRAM[:MEASview<y>] <arg1>
```

```
DISPlay:PULSe:OGRAM[:MEASview<y>]?
```



## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<arg1>= {TVT | FVT | BOTH}

Where

TVT =Time Vs Time

FVT =Frequency Vs Time

BOTH =Both

## Returns

<String> TVT or FVT or BOTH

## Examples

DISPlay:PULSe:OGRAM:MEASview1 FVT sets Frequency vs Time as the display type.

## DISPlay:PULSe:RESult[:MEASview<y>]:ATX

Determines whether or not to show the average transmitted power measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:ATX { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:ATX?
```

## Arguments

OFF or 0 does not show the average transmitted power measurement result.

ON or 1 shows the average transmitted power measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:ATX ON shows the average transmitted power measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:AVERage

Determines whether or not to show the average on power measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:AVERage { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:AVERage?
```

## Arguments

OFF or 0 does not show the average on power measurement result.

ON or 1 shows the average on power measurement result in the results table.

## Returns

See Arguments.

## Examples

```
DISPLAY:PULSE:RESULT:MEASVIEW1:AVERAGE ON shows the average on power measurement result in the pulse table.
```

## DISPlay:PULSe:RESult[:MEASview<y>]:DRODb

Determines whether or not to show the droop measurement result in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:DRODb { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:DRODb?
```

## Arguments

OFF or 0 does not show the droop measurement result in dB.

ON or 1 shows the droop measurement result in dB in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:DRODB ON` shows the droop measurement result in dB in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:DR0op

Sets or queries showing the droop measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:DR0op { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:DR0op?
```

### Arguments

OFF or 0 does not show the droop measurement result.

ON or 1 shows the droop measurement result in the pulse table.

### Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:DROOP ON` shows the droop measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:DUTPct

Determines whether or not to show the duty factor (%) measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:DUTPct { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:DUTPct?
```

### Arguments

OFF or 0 does not show the duty factor measurement result.

ON or 1 shows the duty factor measurement result in the pulse table.

### Returns

See Arguments.

### Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:DUTPCT ON` shows the duty factor (%) measurement result in the pulse table.

## **DISPlay:PULSe:RESult[:MEASview<y>]:DUTRatio**

Determines whether or not to show the duty factor (ratio) measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:DUTRatio { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:DUTRatio?
```

### Arguments

OFF or 0 does not show the duty factor measurement result.

ON or 1 shows the duty factor measurement result in the pulse table.

### Returns

See Arguments.

### Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:DUTRATIO ON` shows the duty factor (ratio) measurement result in the pulse table.

## **DISPlay:PULSe:RESult[:MEASview<y>]:FABS**

Determines whether or not to show the absolute frequency measurement result in the Pulse Table.

### Conditions

Measurement views: Pulse Table

### Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:FABS {OFF | ON | 0 | 1}
```

## Arguments

OFF or 0 does not show the absolute frequency measurement result in the pulse table.

ON or 1 shows the absolute frequency measurement result in the pulse table.

## Returns

See Arguments.

## Examples

```
DISPLAY:PULSE:RESULT:MEASVIEW1:FABS ON shows the absolute frequency measurement result in the pulse table.
```

## DISPlay:PULSe:RESult[:MEASview<y>]:FALL

Determines whether or not to show the fall time measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:FALL { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:FALL?
```

## Arguments

OFF or 0 does not show the fall time measurement result.

ON or 1 shows the fall time measurement result in the pulse table.

## Returns

See Arguments.

## Examples

```
DISPLAY:PULSE:RESULT:MEASVIEW1:FALL ON shows the fall time measurement result in the pulse table.
```

## DISPlay:PULSe:RESult[:MEASview<y>]:FDELta

Determines whether or not to show the delta frequency measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:FDELta { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 does not show the delta frequency measurement result.

ON or 1 shows the delta frequency measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:FDELTA ON shows the delta frequency measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:FRDeviation

Determines whether or not to show the frequency deviation measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:FRDeviation { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:FRDeviation?
```

## Arguments

OFF or 0 does not show the frequency deviation measurement result.

ON or 1 shows the frequency deviation measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:FRDEVIATION ON shows the frequency deviation measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:IRAMplitude

Sets or queries showing the impulse response amplitude measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:IRAMplitude { OFF | ON | 0 | 1 }  
DISPlay:PULSe:RESult[:MEASview<y>]:IRAMplitude?
```

### Arguments

OFF or 0 does not show the Impulse Response Amplitude measurement result.

ON or 1 shows the Impulse Response Amplitude measurement result in the results table.

### Returns

See Arguments.

### Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:IRAMPLITUDE ON shows the Impulse Response Amplitude measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:IRTime

Sets or queries showing the Impulse Response Time measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:IRTime { OFF | ON | 0 | 1 }  
DISPlay:PULSe:RESult[:MEASview<y>]:IRTime?
```

### Arguments

OFF or 0 does not show the Impulse Response Time measurement result.

ON or 1 shows the Impulse Response Time measurement result in the results table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:IRTIME ON` shows the Impulse Response Time measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:MFReqerror

Determines whether or not to show the maximum frequency error measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:MFReqerror { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:MFReqerror?
```

## Arguments

OFF or 0 does not show the maximum frequency error measurement result.

ON or 1 shows the maximum frequency error measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:MFREQERROR ON` shows the maximum frequency error measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:MPHerror

Determines whether or not to show the maximum phase error measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands



## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:MPHerror { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult[:MEASview<y>]:MPHerror?
```

## Arguments

OFF or 0 does not show the maximum phase error measurement result.

ON or 1 shows the maximum phase error measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:MPHERROR ON shows the maximum phase error measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:OVEDb

Determines whether or not to show the Overshoot measurement result in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:OVEDb { OFF | ON | 0 | 1 }
DISPlay:PULSe:RESult[:MEASview<y>]:OVEDb?
```

## Arguments

OFF or 0 does not show the Overshoot measurement result in dB.

ON or 1 shows the Overshoot measurement result in dB in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:OVEDB ON shows the Overshoot measurement result in dB in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:OVERshoot

Sets or queries whether or not to show the Overshoot measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:OVERshoot { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:OVERshoot?
```

## Arguments

OFF or 0 does not show the Overshoot measurement result.

ON or 1 shows the Overshoot measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:OVERSHOOT ON shows the Overshoot measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PHDeviation

Determines whether or not to show the phase deviation measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:PHDeviation { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:PHDeviation?
```

## Arguments

OFF or 0 does not show the phase deviation measurement result.

ON or 1 shows the phase deviation measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:PHDEVIATION ON` shows the phase deviation measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PPPD

Determines whether or not to show the pulse to pulse phase measurement result in the Pulse Table.

### Conditions

Measurement views: Pulse Table

### Group

Display commands

### Syntax

`DISPlay:PULSe:RESult[:MEASview<y>]:PPPD {OFF | ON | 0 | 1}`

### Arguments

OFF or 0 does not show the pulse to pulse phase measurement result in the pulse table.

ON or 1 shows the pulse to pulse phase measurement result in the pulse table.

### Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:PPPD ON` shows the pulse to pulse frequency measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PPFD

Determines whether or not to show the pulse to pulse frequency measurement result in the Pulse Table.

### Conditions

Measurement views: Pulse Table

### Group

Display commands

### Syntax

`DISPlay:PULSe:RESult[:MEASview<y>]:PPFD {OFF | ON | 0 | 1}`

### Arguments

OFF or 0 does not show the pulse to pulse frequency measurement result in the pulse table.

ON or 1 shows the pulse to pulse frequency measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:PPFD ON` shows the pulse to pulse frequency measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PPFREquency

Determines whether or not to show the pulse-pulse carrier frequency measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:PPFREquency { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:PPFREquency?
```

## Arguments

OFF or 0 does not show the pulse-pulse carrier frequency measurement result.

ON or 1 shows the pulse-pulse carrier frequency measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:PPFREQUENCY ON` shows the pulse-pulse carrier frequency measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PPOWer

Determines whether or not to show the peak power measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:PPOWer { OFF | ON | 0 | 1 }
```

---

```
DISPlay:PULSe:RESult[:MEASview<y>]:PPOWer?
```

### Arguments

OFF or 0 does not show the peak power measurement result.

ON or 1 shows the peak power measurement result in the pulse table.

### Returns

See Arguments.

### Examples

```
DISPLAY:PULSE:RESULT:MEASVIEW1:PPOWER ON
```

 shows the peak power measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:PPPHase

Determines whether or not to show the pulse-pulse carrier phase measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult:PPPHase { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:PPPHase?
```

### Arguments

OFF or 0 does not show the pulse-pulse carrier phase measurement result.

ON or 1 shows the pulse-pulse carrier phase measurement result in the pulse table.

### Returns

See Arguments.

### Examples

```
DISPLAY:PULSE:RESULT:MEASview1:PPPHASE ON
```

 shows the pulse-pulse carrier phase measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RINTerval

Determines whether or not to show the repetition interval measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RINTERval { OFF | ON | 0 | 1 }  
DISPlay:PULSe:RESult[:MEASview<y>]:RINTERval?
```

## Arguments

OFF or 0 does not show the repetition interval measurement result.

ON or 1 shows the repetition interval measurement result in the results table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:RINTERVAL ON shows the repetition interval measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RIPDb

Sets or queries showing the ripple measurement result in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RIPDb { OFF | ON | 0 | 1 }  
DISPlay:PULSe:RESult[:MEASview<y>]:RIPDb?
```

## Arguments

OFF or 0 does not show the ripple measurement result in dB.

ON or 1 shows the ripple measurement result in dB in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:RIPDB ON shows the ripple measurement result in dB in the pulse table.

## DISPlay:PULSe:RESult:MEASview<y>:RIPPlE

Sets or queries showing the ripple measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult:MEASview<y>:RIPPlE { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult:MEASview<y>:RIPPlE?
```

### Arguments

OFF or 0 does not show the ripple measurement result.

ON or 1 shows the ripple measurement result in the pulse table.

### Returns

See Arguments.

### Examples

DISPlay:PULSe:RESult:MEASview1:RIPPlE ON shows the ripple measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RISE

Determines whether or not to show the rise time measurement result in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Display commands

### Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RISE { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:RISE?
```

### Arguments

OFF or 0 does not show the rise time measurement result.

ON or 1 shows the rise time measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:RISE ON` shows the rise time measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RMSFreqerror

Determines whether or not to show the RMS frequency error measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RMSFreqerror { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:RMSFreqerror?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional.  
Default value 1.

OFF or 0 does not show the RMS frequency error measurement result.

ON or 1 shows the RMS frequency error measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:RMSFREQERROR ON` shows the RMS frequency error measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RMSPHerror

Determines whether or not to show the RMS phase error measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands



## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RMSPherror { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:RMSPherror?
```

## Arguments

OFF or 0 does not show the RMS phase error measurement result.

ON or 1 shows the RMS phase error measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:RMSPHERROR ON shows the RMS phase error measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:RRATe

Determines whether or not to show the repetition rate measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:RRATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:RRATe?
```

## Arguments

OFF or 0 does not show the repetition rate measurement result.

ON or 1 shows the repetition rate measurement result in the pulse table.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:RESULT:MEASVIEW1:RRATE ON shows the repetition rate measurement result in the pulse table.

## DISPlay:PULSe:RESult[:MEASview<y>]:TIME

Determines whether or not to show the time measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:TIME { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:TIME?
```

## Arguments

OFF or 0 does not show the time measurement result.

ON or 1 shows the time measurement result in the pulse table.

## Returns

See Arguments.

## Examples

```
DISPLAY:PULSE:RESULT:MEASVIEW1:TIME ON shows the time measurement result in the pulse table.
```

## DISPlay:PULSe:RESult[:MEASview<y>]:WIDTh

Determines whether or not to show the pulse width measurement result in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Display commands

## Syntax

```
DISPlay:PULSe:RESult[:MEASview<y>]:WIDTh { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:RESult[:MEASview<y>]:WIDTh?
```

## Arguments

OFF or 0 does not show the pulse width measurement result.

ON or 1 shows the pulse width measurement result in the pulse table.

## Returns

See Arguments.

## Examples

`DISPLAY:PULSE:RESULT:MEASVIEW1:WIDTH ON` shows the pulse width measurement result in the pulse table.

## DISPlay:PULSe[:MEASview<y>]:SElect:NUMBer

Sets or queries a pulse to measure. For the selected pulse, the statistics view indicates the measurement result while the table view highlights it, and the trace view displays the waveform.

### Conditions

Measurement views: : Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Display commands

### Syntax

`DISPlay:PULSe[:MEASview<y>]:SElect:NUMBer <number>`

`DISPlay:PULSe[:MEASview<y>]:SElect:NUMBer?`

### Arguments

`<number>`: : `<NR1>` specifies the number of pulse to measure. Range: -(the number of acquired pulses before the time reference) to +(the number of acquired pulses after the time reference).

Zero (0) represents the pulse at the analysis time reference specified using the `[SENSe]:ANALysis:REFerence` command. The number of acquired pulses depends on the analysis range.

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<NR1>`

## Examples

`DISPlay:PULSe:MEASview2:SElect:NUMBer -28` measures the pulse #-28.

## DISPlay:PULSe[:MEASview<y>]:SElect:RESult

Sets or queries which result is shown in the pulse trace and statistics views.

### Conditions

Measurement views: Pulse statistics, Pulse trace

### Group

Display commands

## Syntax

```
DISPlay:PULSe[:MEASview<y>]:SElect:RESult { AVERAge | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPple | RIPDb | DROop | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFreqerror | MFRequerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude | IRTIME }
```

```
DISPlay:PULSe[:MEASview<y>]:SElect:RESult?
```

## Arguments

The following table lists the arguments.

**Table 39: Pulse results**

Argument	Result
AVERAge	Average on power
PPOWer	Peak power
ATX	Average transmitted power
WIDTH	Pulse width
RISE	Rise time
FALL	Fall time
RINTerval	Repetition interval
RRATe	Repetition rate
DUTPct	Duty factor (%)
DUTRatio	Duty factor (ratio)
RIPple	Ripple
RIPDb	Ripple (dB)
DROop	Droop
DRODb	Droop (dB)
OVERshoot	Overshoot
OVEDb	Overshoot (dB)
PPPHase	Pulse-pulse carrier phase
PPFRequency	Pulse-pulse carrier frequency
RMSFreqerror	RMS frequency error
MFRequerror	Maximum frequency error
RMSPherror	RMS phase error
MPHerror	Maximum phase error
FRDeviation	Frequency deviation
PHDeviation	Phase deviation
FDELta	Delta frequency
IRAMplitude	Impulse Response Amplitude
IRTIME	Impulse Response Sidelobe Offset Time from Mainlobe

## Returns

See Arguments.

## Examples

`DISPlay:PULSe:MEASview1:SElect:RESult AVERAge` shows the average on power result in the pulse trace and statistics views.

## DISPlay:PULSe[:MEASview<y>]:SElect:RESult

Sets or queries the result shown in the pulse trace and statistics views.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

`DISPlay:PULSe[:MEASview<y>]:SElect:RESult <arg>`

`DISPlay:PULSe[:MEASview<y>]:SElect:RESult?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

AVERAge is the average on power

PPOWer is the peak power

ATX is the average transmitted power

WIDTh is the Pulse width

RISE is the rise time

FALL is the fall time

RINterval is the repetition interval

RRATe is the repetition rate

DUTPct is the duty factor (%)

DUTRatio is the duty factor (ratio)

RIPPlE is the ripple

RIPDb is the ripple (dB)

DROop is the droop

DRODb is the droop (dB)

OVERshoot is the overshoot

OVEDb is the overshoot (dB)

PPPHase is the pulse-pulse carrier phase

PPFRrequency is the pulse-pulse carrier frequency

RMSFreqerror is the RMS frequency error

MFRreqerror is the maximum frequency error

RMSPherror is the RMS phase error

MPHerror is the maximum phase error

FRDeviation is the frequency deviation

FDELta is the delta frequency

PHDeviation is the phase deviation

IRAmplitude is the impulse response amplitude

TIME is the time in seconds

IRTime is the impulse response sidelobe offset time from the mainlobe

PPPD is the pulse to pulse phase difference

PPFD is the pulse to pulse frequency difference

FABS is the absolute frequency

## Returns

<NRf>

## Examples

DISPlay:PULSe:MEASview1:SElect:RESult AVERAGE shows the average on power result in the pulse trace and statistics views.

## DISPlay:PULSe:STATistics[:MEASview<y>]:MARKer<x>:SHOW:STATe

Determines whether to show or hide the marker readout in the statistics graph. This command is valid when DISPlay:PULSe:STATistics:PLOT is set to FFT.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:MARKer<x>:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:STATistics[:MEASview<y>]:MARKer<x>:SHOW:STATe?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

OFF or 0 hides the marker readout.

ON or 1 shows the marker readout.

## Returns

See Arguments.

## Examples

DISPlay:PULSe:STATistics:MEASview1:MARKer:SHOW:STATE ON shows the marker readout in the statistics graph.

## DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT

Sets or queries how to show the statistics graph.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT { TRENd | FFT | TTRend | HISTogram }

DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<x> = 1 to 4; MARKer0 (reference marker) is invalid.

TRENd shows the statistics result along with the pulse number.

FFT shows the statistics result transformed into the frequency domain by FFT.

TTRend shows the statistics result along with time.

HISTogram shows the histogram statistics results.

## Returns

See Arguments.

## Examples

DISPLAY:PULSE:STATISTICS:MEASview1:PLOT TRENd shows the statistics result along with the pulse number.

## **DISPlay:PULSe:STATistics[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid in the statistics view.

### **Conditions**

Measurement views: Pulse statistics

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:STATistics[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON  
| 0 | 1 }
```

```
DISPlay:PULSe:STATistics[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

### **Arguments**

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Examples**

DISPLAY:PULSE:STATISTICS:MEASVIEW2:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the Pulse Statistics.

## **DISPlay:PULSe:STATistics[:MEASview<y>]:X:RSCale (No Query Form)**

Rescales the horizontal axis to fit the waveform to the screen in the statistics graph.

### **Conditions**

Measurement views: Pulse statistics

### **Group**

Display commands

### **Syntax**

```
DISPlay:PULSe:STATistics[:MEASview<y>]:X:RSCale
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.



## Examples

`DISPlay:PULSe:STATistics:MEASview2:X:RSCale` rescales the horizontal axis of the statistics graph.

## DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALE]:NUMBER

Sets or queries the horizontal full scale in the statistics graph.

### Conditions

Measurement views: Pulse statistics

### Group

Display commands

### Syntax

`DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALE]:NUMBER <value>`

`DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALE]:NUMBER?`

### Related Commands

`DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALE]:OFFSET`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<value>::<NRf> specifies the horizontal full scale. The setting range depends on the `DISPlay:PULSe:STATistics:PLOT` command parameters as shown in the table below.

DISPlay:PULSe:STATistics:PLOT	Setting range
TREND	1 to 1000
FFT	1 Hz to 120 MHz

### Returns

<NRf>

### Examples

`DISPlay:PULSe:STATistics:MEASview2:X:SCALE:NUMBER 50` sets the horizontal full scale to 50 pulses when the plot is trend.

## DISPlay:PULSe:STATistics[:MEASview<y>]:X:SCALE:OFFSET

Sets or queries the minimum horizontal value (the first pulse to show) in the statistics graph.

### Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:X:SCALE:OFFSet <value>
```

```
DISPlay:PULSe:STATistics[:MEASview<y>]:X:SCALE:OFFSet?
```

## Arguments

<value> : <NRf> specifies the number of the first pulse.

Range:  $-(X - X/10)$  to  $+(X - X/10)$  where X is the horizontal scale set by the `DISPlay:PULSe:STATistics:X[:SCALE]:NUMBER` command.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

```
DISPLAY:PULSE:STATISTICS:MEASview2:X:SCALE:OFFSET 120 sets the first pulse number to #120.
```

## DISPlay:PULSe:STATistics[:MEASview<y>]:Y:RSCale (No Query Form)

Rescales the vertical axis to fit the waveform to the screen in the statistics graph.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y:RSCale
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

```
DISPlay:PULSe:STATistics:MEASview2:Y:RSCale rescales the vertical axis of the statistics graph.
```

## DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:FULL

Sets or queries the vertical full scale in the statistics graph.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:FULL <value>
```

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:FULL
```

## Related Commands

```
DISPlay:PULSe:STATistics[:MEASview<y>]:X[:SCALE]:OFFSET
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<value>::<Nrf> specifies the vertical full scale. The setting range depends on the DISPlay:PULSe:STATistics:PLOT command parameters and the measurement items as shown in the following table.

DISPlay:PULSe:STATistics:PLOT	Measurement item	Setting range
TREND	Average on power, Peak power, Average transmitted power	0.1 to 200 dB
	Pulse width, Rise time, Fall time, Repetition interval	1 n to 5 Ms
	Repetition rate	100 m to 100 MHz
	Duty factor, Ripple	1 to 100%
	Drop	1 to 200%
	Pulse-pulse carrier phase	1 to 360°
	FFT	All

## Returns

<Nrf>

## Examples

DISPlay:PULSe:STATistics:MEASview2:Y:SCALE:FULL 100 sets the vertical full scale to 100 dB.

## DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset in the statistics graph.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

<value> : :<NRF> specifies the vertical offset. The vertical offset is the value at the top or the bottom edge of the graph depending on measurement items. The setting range depends on the DISPlay:PULSe:STATistics:PLOT command parameters and the measurement items as shown in the table below.

DISPlay:PULSe:STATistics:PLOT	Measurement item	Setting range	Offset position <sup>1</sup>
TRENd	Average on power, Peak power, Average transmitted power	-170 to +50 dBm	Top (Bottom for the unit of Volts or Watts)
	Pulse width, Rise time, Fall time, Repetition interval	0 to 5 Ms	Bottom
	Repetition rate	0 to 100 MHz	Bottom
	Duty factor, Ripple	0 to 100%	Bottom
	Droop	0 to +100%	Bottom
	Pulse-pulse carrier phase	-180 to +180°	Bottom
FFT	All	-400 to +100 dB	Top

## Returns

<NRF>

## Examples

DISPLAY:PULSE:STATISTICS:MEASview2:Y:SCALE:OFFSET 24.8 sets the maximum vertical value to 24.8 dBm in the statistics graph.

## DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:STOP? (Query Only)

Queries the minimum vertical value (bottom edge) in the statistics graph.

## Conditions

Measurement views: Pulse statistics

## Group

Display commands

## Syntax

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:STOP?
```

## Related Commands

```
DISPlay:PULSe:STATistics[:MEASview<y>]:Y[:SCALE]:OFFSET
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<y\_stop>::<NRf> is the minimum vertical value (bottom edge).

## Examples

DISPlay:PULSe:STATistics:MEASview2:Y:SCALE:STOP? might return -150.0, indicating that the minimum vertical value is -150 dBm in the pulse statistics graph.

## DISPlay:PULSe:TRACe[:MEASview<y>]:MARKer:SHOW:STATe

Determines whether to show or hide the marker readout in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:MARKer:SHOW:STATe?
```

## Arguments

OFF or 0 hides the marker readout.

ON or 1 shows the marker readout.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

`DISPLAY:PULSE:TRACE:MEASVIEW1:MARKER:SHOW:STATE ON` shows the marker readout on the pulse trace measurement view 1.

## **DISPlay:PULSe:TRACe[:MEASview<y>]:POINT:SHOW**

Determines whether to show or hide the measurement points and lines in the pulse trace view.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Display commands

### Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:POINT:SHOW { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:POINT:SHOW?
```

### Arguments

OFF or 0 hides the measurement points and lines.

ON or 1 shows the measurement points and lines.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

`DISPLAY:PULSE:TRACE:MEASVIEW1:POINT:SHOW ON` shows the measurement points and lines in the pulse trace measurement view 1.

## **DISPlay:PULSe:TRACe[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE**

Determines whether to show or hide the graticule grid in the pulse trace view.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Display commands

### Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE { OFF | ON | 0 | 1 }
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATE?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:WINDOW:TRACE:GRATICULE:GRID:STATE ON shows the graticule grid on the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:X:RSCale

Rescales the horizontal axis to fit the waveform to the screen in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X:RSCale
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:X:RSCALE rescales the horizontal axis in the pulse trace Measurement View 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]

Sets or queries the horizontal full scale in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE] <value>
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]?
```

## Arguments

<value> : : <NRf> specifies the horizontal full scale.

Range: 10 ns to acquisition memory capacity.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:X:SCALE 5.5E-6 sets the horizontal scale to 5.5  $\mu$ s in the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:FULL

Sets or queries the full-scale reference for the horizontal rescale.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:FULL { SELEcted | MAXimum }
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:FULL?
```

## Arguments

SELEcted uses the selected pulse for the full-scale reference.

MAXimum uses the maximum pulse for the full-scale reference.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:X:SCALE:FULL SELEcted uses the selected pulse for the full-scale reference in the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALe]:OFFSet

Sets or queries the minimum horizontal value (left edge) in the pulse trace view.



## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:OFFSet <value>
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:OFFSet?
```

## Related Commands

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:PDIVision
```

## Arguments

<value>::<NRf> specifies the minimum horizontal value.

Range: [(analysis offset) - (X scale) × 0.9] to [(analysis offset) + (analysis length) - (X scale) × 0.1]

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:X:SCALE:OFFSET 937.5E-9 sets the minimum horizontal value to 937.5 ns in the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:PDIVision

Sets or queries the horizontal full scale in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:PDIVision <value>
```

```
DISPlay:PULSe:TRACe[:MEASview<y>]:X[:SCALE]:PDIVision?
```

## Arguments

<value>::<NRf> specifies the horizontal full scale.

Range: 10 ns to acquisition memory capacity.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:X:SCALE:PDIVISION 5.5E-6 sets the horizontal scale to 5.5  $\mu$ s in the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:Y:RSCale

Rescales the vertical axis to fit the waveform to the screen in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

DISPlay:PULSe[:MEASview<y>]:TRACe:Y:RSCale

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:Y:RSCALE rescales the vertical axis in the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALE]:FULL

Sets or queries the vertical full scale in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALE]:FULL <value>

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALE]:FULL?

## Related Commands

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:OFFSet

## Arguments

<value>::<NRf> specifies the vertical full scale.

Range: 0.1 to 200 dB.

The unit can be changed by the [SENSe]:POWer:UNITs command.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:PULSE:TRACE:MEASVIEW1:Y:SCALE:FULL 100 sets the vertical full scale to 100 dB on the pulse trace measurement view 1.

## DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the pulse trace view.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Display commands

## Syntax

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:OFFSet <value>

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:OFFSet?

## Related Commands

DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALe]:STOP?

## Arguments

<value>::<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`DISPLAY:PULSE:TRACE:MEASVIEW1:Y:SCALE:OFFSET 23.5` sets the vertical offset to 23.5 dBm on the pulse trace measurement view 1.

## **DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALE]:STOP**

Queries the minimum vertical value (bottom edge) in the pulse trace view.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

### Group

Display commands

### Syntax

`DISPlay:PULSe:TRACe:MEASview<y>:Y[:SCALE]:STOP?`

### Related Commands

`DISPlay:PULSe:TRACe[:MEASview<y>]:Y[:SCALE]:OFFSet`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<y\_stop>: : <NRf> is the minimum vertical value (bottom edge).

## Examples

`DISPLAY:PULSE:TRACE:MEASVIEW1:Y:SCALE:STOP?` might return -150.0, indicating that the minimum vertical value is -150 dBm in the pulse trace measurement view 1.

## **DISPlay:SEM:MARKer:SHOW:STATe**

Determines whether to show or hide the readout for the selected marker in the Spectral Emissions Mask view.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Display commands

### Syntax

`DISPlay:SEM:MARKer:SHOW:STATe { OFF | ON | 1 | 0 }`

`DISPlay:SEM:MARKer:SHOW:STATe?`

## Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

`DISPLAY:SEM:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

## DISPlay:SEM:RESet:SCALe (No Query Form)

Resets the scale in the Spectral Emissions Mask view.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Display commands

## Syntax

```
DISPlay:SEM:RESet:SCALe
```

## Arguments

None

## Examples

`DISPLAY:SEM:RESET:SCALE` resets the scale in the view.

## DISPlay:SEM:SHOW:LIMit

Sets or queries the appearance of the limits in the Spectral Emissions Mask view

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Display commands

## Syntax

```
DISPlay:SEM:SHOW:LIMit { SHADed | LINE | OFF }
```

```
DISPlay:SEM:SHOW:LIMit?
```

## Arguments

`SHADed` uses shading for the limit area in the display.

`LINE` uses a line for the limit area.

OFF turns off the limits in the display.

## Examples

DISPLAY:SEM:SHOW:LIMIT SHADEd uses shading to identify the limits in the display.

## DISPlay:SEM:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule in the Spectral Emissions Mask view.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Display commands

### Syntax

```
DISPlay:SEM:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }
```

```
DISPlay:SEM:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule in the graph.

ON or 1 shows the graticule in the graph.

## Examples

DISPLAY:SEM:WINDOW:TRACE:GRATICULE:GRID:STATE OFF turns off the graticule in the graph.

## DISPlay:SEM:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal scale automatically to fit the waveform in the Spectral Emissions Mask view.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Display commands

### Syntax

```
DISPlay:SEM:X[:SCALE]:AUTO
```

### Arguments

None

## Examples

DISPLAY:SEM:X[:SCALE]:AUTO rescales the horizontal scale to fit the waveform on the screen.

## DISPlay:SEM:X[:SCALE]:START

Sets or queries the start frequency (left edge) of the Spectral Emissions Mask graph.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Display commands

### Syntax

```
DISPlay:SEM:X[:SCALE]:START <value>
```

```
DISPlay:SEM:X[:SCALE]:START?
```

### Arguments

<value>::<NRf> specifies the start frequency.

## Examples

DISPLAY:SEM:X[:SCALE]:START 10Hz sets the start frequency to 10 Hz in the graph.

## DISPlay:SEM:X[:SCALE]:STOP

Sets or queries the stop frequency (right edge) of the Spectral Emissions Mask graph.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Display commands

### Syntax

```
DISPlay:SEM:X[:SCALE]:STOP <value>
```

```
DISPlay:SEM:X[:SCALE]:STOP?
```

### Arguments

<value>::<NRf> specifies the start frequency.

## Examples

DISPLAY:SEM:X[:SCALE]:STOP 2GHz sets the stop frequency to 2 GHz in the graph.

## **DISPlay:SEM:Y[:SCALE]**

Sets or queries the vertical range of the Spectral Emissions Mask view.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Display commands

### **Syntax**

```
DISPlay:SEM:Y[:SCALE] <value>
```

```
DISPlay:SEM:Y[:SCALE]?
```

### **Arguments**

<value>::= <NRf> specifies the vertical scale.

### **Examples**

```
DISPLAY:SEM:Y[:SCALE] 200 sets the vertical scale to 200 MHz.
```

## **DISPlay:SEM:Y[:SCALE]:AUTO (No Query Form)**

Rescales the vertical scale automatically to fit the waveform in the Spectral Emissions Mask view.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Display commands

### **Syntax**

```
DISPlay:SEM:Y[:SCALE]:AUTO
```

### **Arguments**

None

### **Examples**

```
DISPLAY:SEM:Y[:SCALE]:AUTO rescales the vertical scale to fit the waveform on the screen.
```

## **DISPlay:SEM:Y[:SCALE]:OFFSet**

Sets or queries the vertical offset in the Spectral Emissions Mask view.



## Conditions

Measurement views: Spectral Emissions Mask

## Group

Display commands

## Syntax

```
DISPlay:SEM:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:SEM:Y[:SCALE]:OFFSet?
```

## Arguments

<value> ::= <NRF> specifies the horizontal range.

## Examples

DISPLAY:SEM:Y[:SCALE]:OFFSET 40 sets the vertical position to 40 dBm.

## DISPlay:SGRam[:MEASview<y>]:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the spectrogram view.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:FREQuency:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

DISPlay:SGRam:MEASview1:FREQuency:AUTO rescales the horizontal scale automatically to fit the waveform to the screen in the spectrogram measurement view 1.

## DISPlay:SGRam[:MEASview<y>]:FREQuency:OFFSet

Sets or queries the frequency offset (the value at the center of the horizontal axis) in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:FREQuency:OFFSet <value>
```

```
DISPlay:SGRam[:MEASview<y>]:FREQuency:OFFSet?
```

## Related Commands

```
DISPlay:SGRam[:MEASview<y>]:FREQuency[:SCAlE]
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the frequency offset.

Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

## Examples

```
DISPlay:SGRam:MEASview1:FREQuency:OFFSet 1.45GHz sets the frequency offset to 1.45 GHz in the spectrogram measurement view 1.
```

## DISPlay:SGRam[:MEASview<y>]:FREQuency[:SCAlE]

Sets or queries the horizontal range of the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:FREQuency[:SCAlE] <value>
```

```
DISPlay:SGRam[:MEASview<y>]:FREQuency[:SCAlE]?
```

## Related Commands

```
DISPlay:SGRam[:MEASview<y>]:FREQuency:OFFSet
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the horizontal range.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Returns

<NRf> Returns the horizontal range

## Examples

DISPlay:SGRam:MEASview1:FREQuency:SCAle 10MHz sets the horizontal range to 10 MHz on the trace of Spectrogram measurement view 1.

## DISPlay:SGRam[:MEASview<y>]:MARKer:SHOW:STATe

Sets or queries the display of the Marker legend in the Spectrogram graph.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:MARKer:SHOW:STATe{ OFF | ON | 0 | 1 }
```

```
DISPlay:SGRam[:MEASview<y>]:MARKer:SHOW:STATe?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 removes or turns off the Marker legend.

ON or 1 displays or turns on the Marker legend.

## Returns

```
{ OFF | ON | 0 | 1 }
```

## Examples

DISPlay:SGRam:MEASview1:MARKer:SHOW:STATe ON turns on the display of the Marker legend in the spectrogram graph of view 1.

## DISPlay:SGRam[:MEASview<y>]:SELected:TIMestamp

Sets or queries the display of the time stamp in the Spectrogram graph.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:SElected:TIMEstamp {OFF | ON | 0 | 1}
```

```
DISPlay:SGRam[:MEASview<y>]:SElected:TIMEstamp?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 removes or turns off the time stamp.

ON or 1 displays or turns on the time stamp.

## Returns

```
{OFF | ON | 0 | 1}
```

## Examples

DISPlay:SGRam:MEASview1:SElected:TIMEstamp? queries the display status of the time stamp in the spectrogram graph view 1.

## DISPlay:SGRam[:MEASview<y>]:TIME:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the graph to the screen in the spectrogram view.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

DISPlay:SGRam:MEASview1:TIME:AUTO rescales the vertical scale automatically to fit the waveform to the screen in the spectrogram measurement view 1.

## DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet

Sets or queries the vertical axis (time) offset (bottom line number) in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Display commands

### Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet <value>
```

```
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet?
```

### Related Commands

```
DISPlay:SGRam[:MEASview<y>]:TIME[:SCAlE]
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the time offset.

Range: Line #0 to 125000. Zero (0) represents the latest line.

### Returns

<NRf> specifies the time offset

### Examples

```
DISPlay:SGRam:MEASview1:TIME:OFFSet 15 sets the time offset to Line #15 in the spectrogram view 1.
```

## DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet:DIVisions

Sets or queries the Vertical offset or Position in divisions.

### Conditions

Measurement views: Spectrogram

### Group

Display commands

### Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet:DIVisions <value>
```

```
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet:DIVisions?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NRf> sets the vertical offset value in divisions. Range is -130,000 to +130,000.

## Returns

<NRf> specifies the vertical offset

## Examples

DISPlay:SGRam:MEASview1:TIME:OFFSet:DIVisions 10 sets the vertical offset to 10 divisions in the spectrogram view 1.

## DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap

Sets or queries if overlap is enabled or not.

## Conditions

Measurement view: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap { OFF | ON | 0 | 1 }
```

```
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 hides the overlap in the spectrogram view.

ON or 1 shows the overlap in the spectrogram view.

## Returns

{ OFF | ON | 0 | 1 }

## Examples

DISPlay:SGRam:MEASview1:TIME:OVERlap ON enables time overlap for spectrogram.

## DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent

Sets or queries the Overlap in percent. Note that changing this value may change the time per division DISPlay:SGRam:MEASview<y>:TIME:SCAle:PER:DIVision value.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent <value>
```

```
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : :<NRf> specifies the overlap in percent.

## Returns

<NRf> specifies the overlap in percent

## Examples

```
DISPlay:SGRam:MEASview1:TIME:OVERlap:PERCent 1.0E1 sets the overlap to 10 % in the spectrogram view 1.
```

## DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]



**Note:** *The user interface on the instrument no longer presents this setting. However, the command is still function. Setting this value will change the TIME:PER:DIVision value.*

Sets or queries the vertical scale (the amount of time in each line) in the spectrogram. The vertical axis is composed of successive spectral displays. The new spectra can be added at a timed rate specified by this command. For example, if you set the scale to -5, one line is displayed every 5 spectra.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE] <value>
```

```
DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]?
```

## Related Commands

```
DISPlay:SGRam[:MEASview<y>]:TIME:OFFSet
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NR1> specifies the vertical scale.

Range: -1023 to 0. Zero (0) displays every spectrum.

## Returns

<NRf> specifies the vertical scale

## Examples

DISPlay:SGRam:MEASview1:TIME:SCALE -5 displays one line every 5 spectra in the spectrogram view 1.

## DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision

Sets or queries the vertical Time/division value. Note that changing this value may change the DISPlay:SGRam:MEASview<y>:TIME:OVERlap:PERCent value.

## Conditions

Measurement views: Spectrogram

## Group

Display commands

## Syntax

DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision <value>

DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision?

## Related Commands

DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NRf> specifies the time per division in seconds.

## Returns

<NRf> specifies the time per division in seconds

## Examples

DISPlay:SGRam:MEASview1:TIME:SCALE:PER:DIVision? might return 1.116000000E-3 in the spectrogram view 1.



## DISPlay:SGRam[:MEASview<y>]:TIME:SPECTrums:PERLine (Query Only)

Queries the Spectrums/Line value.

### Conditions

Measurement views: Spectrogram

### Group

Display commands

### Syntax

```
DISPlay:SGRam[:MEASview<y>]:TIME:SPECTrums:PERLine?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

<NRf> specifies the specturm/Line value

### Examples

DISPlay:SGRam:MEASview1:TIME:SPECTrums:PERLine? might return a 1.0000000000 as a response in the spectrogram view 1.

## DISPlay:SPECTrum[:MEASview<y>]:FREQuency:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the spectrum view.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency:AUTO
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

DISPlay:SPECTrum:MEASview2:FREQuency:AUTO rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:SPECTrum[:MEASview<y>]:FREQuency:OFFSet

Sets or queries the frequency offset (the value at the center of the horizontal axis) in the spectrum graph.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency:OFFSet <value>
```

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency:OFFSet?
```

### Related Commands

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency:OFFSet[:SCALE]
```

### Arguments

<value>::<NRf> specifies the frequency offset.

Range: [(center frequency) - (X scale) × 0.9] to [(center frequency) + (X scale) × 0.9]

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

DISPlay:SPECTrum:MEASview2:FREQuency:OFFSet 1.45GHz sets the frequency offset to 1.45 GHz in the spectrum.

## DISPlay:SPECTrum[:MEASview<y>]:FREQuency[:SCALE]

Sets or queries the horizontal range of the spectrum graph.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency[:SCALE] <value>
```

```
DISPlay:SPECTrum[:MEASview<y>]:FREQuency[:SCALE]?
```

### Related Commands

[DISPlay:SPECTrum\[:MEASview<y>\]:FREQuency:OFFSet](#)

## Arguments

<value> : : <NRf> specifies the horizontal range.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

DISPlay:SPECTrum:MEASview2:FREQuency:SCAle 10MHz sets the horizontal range to 10 MHz.

## DISPlay:SPECTrum[:MEASview<y>]:MARKer:NOISe:MODE

Determines whether to enable or disable the marker noise mode in the spectrum view. In this mode, the marker readout indicates amplitude in dBm/Hz. It is valid for all markers except for the reference marker.



**Note:** To use the marker noise mode, select dBm as the power unit by the [SENSe]:POWer:UNITs command.



**Note:** It is an overlapped command, which does not finish executing before the next command starts executing. Use the \*OPC(?) command to synchronize all pending operations to the execution of this command.

## Conditions

Measurement views: Spectrum

## Group

Display commands

## Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:MARKer:NOISe:MODE { OFF | ON | 0 | 1 }
```

```
DISPlay:SPECTrum[:MEASview<y>]:MARKer:NOISe:MODE?
```

## Related Commands

[\[SENSe\]:POWer:UNITs](#)

## Arguments

OFF or 0 disables the marker noise mode.

ON or 1 enables the marker noise mode.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

`DISPlay:SPECTrum:MEASview2:MARKer:NOISe:MODE ON` enables the marker noise mode

## **DISPlay:SPECTrum[:MEASview<y>][:SCALE]:LOG:STATe**

Determines whether or not to set the horizontal axis logarithmic in the Spectrum view.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>][:SCALE]:LOG:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPECTrum[:MEASview<y>][:SCALE]:LOG:STATe?
```

### Related Commands

```
DISPlay:SPECTrum:X:LABel
```

### Arguments

OFF or 0 sets the horizontal axis linear (default).

ON or 1 sets the horizontal axis logarithmic.

Executing `DISPlay:SPECTrum:SCALE:LOG:STATe ON` sets `DISPlay:SPECTrum:X:LABel SSFReq`.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`DISPlay:SPECTrum:MEASview2:SCALE:LOG:STATe ON` sets the horizontal axis logarithmic in the Spectrum view.

## **DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

```
DISPlay:SPECTrum:MEASview2:WINDow:TRACe:GRATicule:GRID:STATe ON shows the graticule grid on the screen in the spectrum measurement.
```

## DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:LEGend:STATe

Determines whether to show or hide the trace legend in the Spectrum view. The legend indicates the trace detection and function on the screen for each displayed spectrum trace.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPECTrum[:MEASview<y>]:WINDow:TRACe:LEGend:STATe?
```

### Arguments

OFF or 0 hides the trace legend.

ON or 1 shows the trace legend.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

```
DISPlay:SPECTrum:MEASview2:WINDow:TRACe:LEGend:STATe ON shows the trace legend on the screen in the spectrum measurement.
```

## DISPlay:SPECTrum[:MEASview<y>]:X:LABel

Sets or queries the labels for the horizontal (X) axis in the Spectrum view. The labels are indicated beneath the spectrum graph on the screen.

### Conditions

Measurement views: Spectrum

## Group

Display commands

## Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:X:LABel { SSFReq | CFSPan }
```

```
DISPlay:SPECTrum[:MEASview<y>]:X:LABel?
```

## Arguments

`SSFReq` sets the labels to the start and stop frequencies.

`CFSPan` sets the labels to the center frequency and span.

The parameter `<y>`=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Examples

`DISPlay:SPECTrum:MEASview2:X:LABel SSFReq` sets the labels to the start and stop frequencies for the horizontal axis in the Spectrum view.

## DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]

Sets or queries the vertical range of the spectrum graph.

## Conditions

Measurement views: Spectrum

## Group

Display commands

## Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE] <value>
```

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]?
```

## Arguments

`<value>`: : <NRf> specifies the vertical range. Range: 0.1 to 200 dB. The amplitude unit is set by the [\[SENSe\]:POWer:UNITs](#) command.

You can omit the unit in the argument. When you include the unit in the argument, only dB is available. For the amplitude units Watts, Volts, and Amps, omit the unit in the argument.

The parameter `<y>`=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Examples

`DISPlay:SPECTrum:MEASview2:Y:SCALE 50dB` sets the vertical range to 50 dB in the Spectrum view.

## DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Spectrum view.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]:AUTO
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

DISPlay:SPECTrum:MEASview2:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen

## DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]:OFFSET

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the spectrum graph.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]:OFFSET <value>
```

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCALE]:OFFSET?
```

### Arguments

<value>::<NRf> specifies the vertical offset. Range: -270 to +150 dBm. The amplitude unit is set by the [\[SENSe\]:POWer:UNITs](#) command (default: dBm).

You can omit the unit in the argument. When you include the unit in the argument, only dBm is allowed. For the other amplitude units, omit the unit in the argument.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

DISPlay:SPECTrum:MEASview2:Y:SCALE:OFFSET -12.5 sets the vertical offset to -12.5 dBm

## DISPlay:SPECTrum[:MEASview<y>]:Y[:SCAlE]:PDIVision

Sets or queries the vertical scale (per division) of the spectrum graph.

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCAlE]:PDIVision <value>
```

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCAlE]:PDIVision?
```

### Related Commands

[\[SENSe\]:POWer:UNITs](#)

### Arguments

<value>::<NRf> specifies the vertical scale (per division).

Range: 0.01 to 20 dB/div.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

```
DISPlay:SPECTrum:MEASview2:Y:SCAlE:PDIVision 0.5 sets the vertical scale to 0.5 dB/div.
```

## DISPlay:SPECTrum[:MEASview<y>]:Y[:SCAlE]:RESet (No Query Form)

Resets the vertical scale of the spectrum graph to the default values: Vertical offset = Reference level and Vertical scale = 100 dB

### Conditions

Measurement views: Spectrum

### Group

Display commands

### Syntax

```
DISPlay:SPECTrum[:MEASview<y>]:Y[:SCAlE]:RESet
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1



## Examples

`DISPlay:SPECTrum:MEASview2:Y:SCAle:RESet` resets the vertical scale to the default values in the Spectrum view

## DISPlay:SPURious:MARKer:SHOW:STATe

Determines whether to show or hide the readout for the selected marker in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:MARKer:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPURious:MARKer:SHOW:STATe?
```

### Arguments

OFF or 0 hides the readout for the selected marker in the graph.

ON or 1 shows the readout for the selected marker in the graph.

## Examples

`DISPLAY:SPURIOUS:MARKER:SHOW:STATE ON` shows the readout for the selected marker in the graph.

## DISPlay:SPURious:RESet:SCALe (No Query Form)

Resets the horizontal and vertical scale to the default values described below in the Spurious view.

Vertical offset = 0 dBm, Vertical scale = 100 dB,

Horizontal offset = Center frequency, and Horizontal scale = Default span

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:RESet:SCALe
```

### Arguments

None

## Examples

`DISPLAY:SPURIOUS:RESET:SCALE` resets the horizontal and vertical scale to the default values.

## DISPlay:SPURious:SCALe:LOG:STATe

Determines whether or not to set the horizontal axis logarithmic in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:SCALe:LOG:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPURious:SCALe:LOG:STATe?
```

### Arguments

OFF or 0 sets the horizontal axis linear (default).

ON or 1 sets the horizontal axis logarithmic.

## Examples

`DISPLAY:SPURIOUS:SCALE:LOG:STATE ON` sets the horizontal axis logarithmic in the Spurious view.

## DISPlay:SPURious:SELEct:NUMBer

Sets or queries the spurious number in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:SELEct:NUMBer <number>
```

```
DISPlay:SPURious:SELEct:NUMBer?
```

### Arguments

`<number>::<NR1>` specifies the spurious number.

Range: 1 to the number of spurious signals.

Use the [FETCh:SPURious:COUNT?](#) or [READ:SPURious:COUNT?](#) query to get the number of spurious signals.

## Examples

`DISPLAY:SPURIOUS:SELECT:NUMBER 7` selects the spurious #7.

## DISPlay:SPURious:SHOW:LIMit

Sets or queries how to display the limits.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:SHOW:LIMit { SHADed | LINE | OFF }
```

```
DISPlay:SPURious:SHOW:LIMit?
```

### Arguments

`SHADed` displays the limits with shade.

`LINE` displays the limits with line only.

`OFF` hides the limits.

## Examples

`DISPLAY:SPURIOUS:SHOW:LIMIT LINE` displays the limits with line only.

## DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe

Determines whether to show or hide the graticule grid on the screen.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:SPURious:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

`OFF` or `0` hides the graticule grid.

`ON` or `1` shows the graticule grid.

## Examples

`DISPLAY:SPURIOUS:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen.

## DISPlay:SPURious:X[:SCALE]:AUTO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:X[:SCALE]:AUTO
```

### Arguments

None

## Examples

`DISPLAY:SPURIOUS:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:SPURious:X[:SCALE]:START

Sets or queries the minimum horizontal value (left edge) of the spectrum graph in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

```
DISPlay:SPURious:X[:SCALE]:START <value>
```

```
DISPlay:SPURious:X[:SCALE]:START?
```

### Arguments

`<value>::<NRf>` specifies the minimum horizontal value.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`DISPLAY:SPURIOUS:X:SCALE:START 1.61GHz` sets the minimum horizontal value to 1.61 GHz in the spectrum graph.

## DISPlay:SPURious:X[:SCALE]:STOP

Sets or queries the maximum horizontal value (right edge) of the spectrum graph in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

`DISPlay:SPURious:X[:SCALE]:STOP <value>`

`DISPlay:SPURious:X[:SCALE]:STOP?`

### Arguments

`<value>::<NRF>` specifies the minimum horizontal value.

Range:

- RSA5103AB – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`DISPLAY:SPURIOUS:X:SCALE:STOP 2.16GHz` sets the maximum horizontal value to 2.16 GHz in the spectrum graph.

## DISPlay:SPURious:Y[:SCALE]

Sets or queries the vertical range of the spectrum graph in the Spurious view.

### Conditions

Measurement views: Spurious

### Group

Display commands

### Syntax

`DISPlay:SPURious:Y[:SCALE] <value>`

`DISPlay:SPURious:Y[:SCALE]?`

## Arguments

<value>::<Nrf> specifies the vertical range. Range: 0.1 to 200 dB. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

DISPLAY:SPURIOUS:Y:SCALE 100 sets the vertical range to 100 dB in the Spurious view.

## DISPlay:SPURious:Y[:SCALE]:AUTO (No Query Form)

Rescales the vertical axis automatically to fit the waveform to the screen in the Spurious view.

## Conditions

Measurement views: Spurious

## Group

Display commands

## Syntax

DISPlay:SPURious:Y[:SCALE]:AUTO

## Arguments

None

## Examples

DISPLAY:SPURIOUS:Y:SCALE:AUTO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:SPURious:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) of the spectrum graph in the Spurious view.

## Conditions

Measurement views: Spurious

## Group

Display commands

## Syntax

DISPlay:SPURious:Y[:SCALE]:OFFSet <value>

DISPlay:SPURious:Y[:SCALE]:OFFSet?

## Arguments

<value>::<Nrf> specifies the vertical offset. Range: -270 to +150 dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

`DISPLAY:SPURIOUS:Y:SCALE:OFFSET -12.5` sets the vertical offset to -12.5 dBm in the spectrum graph.

## **DISPlay:TDIagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen in the Trellis diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Display commands

## Syntax

```
DISPlay:TDIagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TDIagram[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Returns

OFF or 0

ON or 1

## Examples

`DISPlay:TDIagram:MEASview2:WINDow:TRACe:GRATICULE:GRID:STATe ON` shows the graticule grid on the screen in the Trellis diagram.

## **DISPlay:TDIagram[:MEASview<y>]:Y[:SCALe]**

Sets or queries the vertical range of the trellis diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Display commands

## Syntax

```
DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE] <value>
```

```
DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]?
```

## Arguments

<value> : <NRf> specifies the vertical range.

Range: 1 ° to 1 T°.

## Returns

<NRf>

## Examples

DISPlay:TDIagram:MEASview2:Y:SCALE 2.5 sets the vertical range to 2.5 ° in the Trellis diagram.

## DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]:AUTO (No query form)

Sets the vertical scale automatically to fit the waveform to the screen in the Trellis diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Display commands

## Syntax

```
DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPlay:TDIagram:MEASview2:Y:SCALE:AUTO sets the vertical scale automatically to fit the waveform to the screen.



## DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (center point of the vertical axis) of the Trellis diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

### Group

Display commands

### Syntax

```
DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:TDIagram[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

### Arguments

<value>::<NRf> specifies the vertical offset.

Range: -1 T° to +1 T°.

### Returns

<NRf>

### Examples

DISPlay:TDIagram:MEASview2:Y:SCALE:OFFSet -28.5 sets the vertical offset to -28.5 ° in the Trellis diagram measview 2.

## DISPlay:TG:MEASview:DELeTe TXGain (No Query Form)

This command removes the Transmission Gain display icon from the Selected displays filed of the Selected Displays window.

### Conditions

Measurement view: Select Displays window

### Group

Display commands

### Syntax

```
DISPlay:TG:MEASview:DELeTe TXGain
```

### Arguments

## Examples

```
DISPLAY:TG:MEASVIEW:DELETE TXGAIN
```

## DISPlay:TG:MEASview:NEW TXGain (No Query Form)

This command opens a new Transmission Gain measurement display window.

### Conditions

Measurement view: Select Displays window

### Group

Display commands

### Syntax

```
DISPlay:TG:MEASview:NEW TXGain
```

## DISPlay:TG:MEASview:SElect TXGain

This command selects the Transmission Gain display icon in the Select Display window and queries if the display is selected or not.

### Conditions

Measurement view: Select Displays window

### Group

Display commands

### Syntax

```
DISPlay:TG:MEASview:SElect TXGain { OFF | ON | 0 | 1 }
```

```
DISPlay:TG:MEASview:SElect TXGain?
```

### Arguments

OFF or 0 deselects the Transmission Gain display icon.

ON or 1 selects the Transmission Gain display icon.

### Returns

0 means the icon is not selected.

1 means the icon is selected.

## Examples

```
DISPLAY:TG:MEASVIEW:SELECT TXGAIN
```

## DISPlay:TOVerview[:MEASview<y>]:WINDow:NAVigator:STATe

Sets or queries whether the navigator view of the Time Overview display is on or off.

### Conditions

Measurement view: Time Overview with specified Measurement view number

### Group

Display commands

### Syntax

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:NAVigator:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:NAVigator:STATe?
```

### Arguments

OFF or 0 sets the Time Overview display to not show in Navigator View mode.

ON or 1 sets the Time Overview display to show in Navigator View mode.

### Returns

OFF or 0 means that the Time Overview display is not in Navigator View mode.

ON or 1 means that the Time Overview display is in Navigator View mode.

### Examples

DISPLAY:TOVERVIEW:MEASVIEW1:WINDOW:NAVIGATOR:STATE ON will set the Time Overview display to show in Navigator View mode.

## DISPlay:TOVerview[:MEASview<y>]:WINDow:TIME:MODE

Sets or queries the type of time analysis to be performed for the Time Overview display.

### Conditions

Measurement view: Time Overview with specified Measurement view number

### Group

Display commands

### Syntax

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TIME:MODE { ANALYsis | SPECTrum | LINKed }
```

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TIME:MODE?
```

### Related Commands

## Arguments

**ANALYSIS** sets the type of analysis time to Analysis length, which is the period of time within the acquisition record over which all other measurements (such as Amplitude vs. Time) are made.

**SPECTRUM** sets the type of analysis time to Spectrum length, which is the period of time within the acquisition record over which the spectrum is calculated.

**LINKED** sets the type of analysis time to lock the Spectrum length and Analysis length together so that the data used to produce the Spectrum display is also used for measurement displays.

## Returns

**ANAL** means that the type of time analysis is set to Analysis.

**SPEC** means that the type of time analysis is set to Spectrum.

**LINK** means that the type of time analysis is set to Linked.

## Examples

`DISPLAY:TOVERVIEW:WINDOW:TIME:MODE?` might return **SPEC**, which indicates that the type of analysis time is set to Spectrum, which is the period of time within the acquisition record over which the spectrum is calculated.

## **DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule grid on the screen.

## Conditions

Measurement views: Time overview

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule grid.

ON or 1 shows the graticule grid.

## Examples

`DISPLAY:TOVERVIEW:MEASVIEW1:WINDOW:TRACE:GRATICULE:GRID:STATE ON` shows the graticule grid on the screen in the time overview Measurement View 1.

## **DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:LEGend:STATe**

Sets or queries the trace legend state in the Time Overview display to on (showing) or off (hidden).

## Conditions

Measurement view: Time Overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:LEGend:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TOVerview[:MEASview<y>]:WINDow:TRACe:LEGend:STATe?
```

## Related Commands

## Arguments

OFF or 0 hides the trace legend in the Time Overview display.

ON or 1 shows the trace legend in the Time Overview display.

## Returns

0 means the trace legend in the Time Overview display is hidden (off).

1 means the trace legend in the Time Overview display is showing (on).

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:WINDOW:TRACE:LEGEND:STATE ON will turn on the trace legend in the Time Overview display.

DISPLAY:TOVERVIEW:MEASVIEW1:WINDOW:TRACE:LEGEND:STATE? 1 indicates that the trace legend in the Time Overview display is showing (on).

## DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]

Sets or queries the horizontal scale (full-scale time) of the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE] <value>
```

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]?
```

## Related Commands

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:OFFSet
```

## Arguments

<value> : : <Nrf> specifies the horizontal scale in full-scale time.

Range: 10 ns to the acquisition memory capacity.

You can see the acquisition memory capacity using the [\[SENSe\]:ACQuisition:MEMory:CAPacity\[:TIME\]?](#) query.

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:X:SCALE 12.5us sets the horizontal scale to 12.5  $\mu$ s.

## DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:AUTO (No Query Form)

Sets the horizontal scale and offset automatically to fit the waveform to the screen in the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:AUTO
```

## Arguments

None

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:X:SCALE:AUTO sets the horizontal scale and offset automatically to fit the waveform on the trace of Time overview Measurement View 1.

## DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:OFFSet

Sets or queries the minimum horizontal value (left edge) of the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:OFFSet <value>
```

```
DISPlay:TOVerview[:MEASview<y>]:X[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:TOVerview\[:MEASview<y>\]:X\[:SCALE\]](#)

## Arguments

<value>::<NRf> specifies the minimum horizontal value.

Range: [(analysis offset) - (X scale) × 0.9] to [(analysis offset) + (analysis length) - (X scale) × 0.1]

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:X:SCALE:OFFSET 800ns sets the minimum horizontal value to 800 ns in the Time overview Measurement View 1.

## DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]

Sets or queries the vertical range of the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE] <value>

DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]?

## Related Commands

[DISPlay:TOVerview\[:MEASview<y>\]:Y\[:SCALE\]:OFFSet](#)

## Arguments

<value>::<NRf> specifies the vertical range. Range: 0.1 to 200 dB.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:Y:SCALE 50 sets the vertical range to 50 dBm in the time overview.

## DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]:AUTO (No Query Form)

Sets the vertical scale and offset automatically to fit the waveform to the screen in the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]:AUTO
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:Y:SCALE:AUTO sets the vertical scale and offset automatically to fit the waveform to the screen.

## DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]:OFFSet

Sets or queries the vertical offset (the value at the top edge of the vertical axis) in the time overview.

## Conditions

Measurement views: Time overview with specified Measurement view number

## Group

Display commands

## Syntax

```
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]:OFFSet <value>
```

```
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALE]:OFFSet?
```

## Related Commands

[DISPlay:TOVerview\[:MEASview<y>\]:Y\[:SCALE\]](#)

## Arguments

<value>::<NRf> specifies the vertical offset. Range: -170 to +50 dBm.

## Returns

<NRf>

## Examples

DISPLAY:TOVERVIEW:MEASVIEW1:Y:SCALE:OFFSET -80 sets the vertical offset to -80 dBm in the time overview Measurement View 1.



## DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]:RESCale (No Query Form)

Sets the vertical scale automatically to fit the waveform to the screen in the time overview.

### Conditions

Measurement views: Time overview with specified Measurement view number

### Group

Display commands

### Syntax

```
DISPlay:TOVerview[:MEASview<y>]:Y[:SCALe]:RESCale
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Examples

DISPlay:TOVERVIEW:MEASVIEW1:Y:SCALE:RESCALE sets the vertical scale automatically to fit the waveform to the screen.

## DISPlay:TXGain:MARKer:SHOW:STATE

Displays or hides the marker readout, but not the marker itself, on the graph area.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:MARKer:SHOW:STATE{ OFF | ON | 0 | 1 }
```

```
DISPlay:TXGain:MARKer:SHOW:STATE?
```

### Arguments

OFF or 0 hides the readout for the selected marker.

ON or 1 shows the readout for the selected marker.

### Returns

OFF or 0 means that the readout for the selected marker is hidden.

ON or 1 means that the readout for the selected marker is showing.

## Examples

`DISPLAY:TXGAIN:MARKER:SHOW:STATE ON` shows the readout for the marker on the display.

## DISPlay:TXGain[:SCALE]:AUTO (No Query Form)

Automatically rescales the horizontal and vertical scale.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain[:SCALE]:AUTO
```

### Arguments

None

## Examples

`DISPLAY:TXGAIN[:SCALE]:AUTO` rescales the horizontal and vertical scale automatically.

## DISPlay:TXGain:WINDow:TRACe:GRATicule:GRID:STATe

Displays or hides the graticule grid in the display. Queries whether the graticule grid is hidden or showing.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TXGain:WINDow:TRACe:GRATicule:GRID:STATe?
```

### Arguments

OFF or 0 hides the graticule.

ON or 1 shows the graticule.

### Returns

OFF or 0 means that the graticule is hidden.

ON or 1 means that the graticule is showing.

## Examples

`DISPLAY:TXGAIN:WINDOW:TRACE:GRATICULE:GRID:STATE OFF` will hide the graticule.

## DISPlay:TXGain:WINDow:TRACe:POINTs:SHOW:STATe

Sets to mark each measurement point on the trace.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:WINDow:TRACe:POINTs:SHOW:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TXGain:WINDow:TRACe:POINTs:SHOW:STATe?
```

### Arguments

OFF or 0 hides each measurement point on the trace.

ON or 1 shows each measurement point on the trace.

### Returns

OFF or 0 means that each measurement point on the trace is hidden.

ON or 1 means that each measurement point on the trace is showing.

## Examples

`DISPLAY:TXGAIN:WINDOW:TRACE:POINTS:SHOW:STATE 1` shows each measurement point on the trace.

## DISPlay:TXGain:X[:SCALE]:AUTO (No Query Form)

Resets the scale of the horizontal axis to contain the complete trace.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:X[:SCALE]:AUTO
```

### Arguments

None

## Examples

`DISPLAY:TXGAIN:X:SCALE:AUTO` resets the scale of the horizontal axis to contain the complete trace.

## DISPlay:TXGain:X[:SCALE]:LOG:STATe

Sets or queries the display to show the frequency axis in a logarithmic scale.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:X[:SCALE]:LOG:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:TXGain:X[:SCALE]:LOG:STATe?
```

### Arguments

OFF or 0 sets the frequency axis in linear scale.

ON or 1 sets the frequency axis in logarithmic scale.

### Returns

OFF or 0 means that the frequency axis is set to linear scale.

ON or 1 means that the frequency axis is set to logarithmic scale.

## Examples

`DISPLAY:TXGAIN:X:SCALE:LOG:STATE1` sets the frequency axis to logarithmic scale.

## DISPlay:TXGain:X[:SCALE]:START

Sets the Zoom Start frequency. This only affects the start frequency shown on the graph.

### Conditions

Measurement view: Transmission Gain

### Group

Display commands

### Syntax

```
DISPlay:TXGain:X[:SCALE]:START <value>
```

```
DISPlay:TXGain:X[:SCALE]:START?
```

## Arguments

`<value> ::= <NRf>` is a floating point number that is the zoom start frequency.

## Returns

See Arguments.

## Examples

`DISPLAY:TXGAIN:X:SCALE:START 1E9` sets the zoom start frequency to 1 GHz.

## DISPlay:TXGain:X[:SCALE]:STOP

Sets the Zoom Stop frequency. This only affects the stop frequency shown on the graph.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

`DISPlay:TXGain:X[:SCALE]:STOP <value>`

`DISPlay:TXGain:X[:SCALE]:STOP?`

## Arguments

`<value> ::= <NRf>` specifies the zoom stop frequency.

## Returns

A floating point value that is the zoom stop frequency.

## Examples

`DISPLAY:TXGAIN:X:SCALE:STOP 1E9` sets the zoom stop frequency to 1 GHz.

## DISPlay:TXGain:Y[:SCALE]

Sets or queries the vertical scale value in dB.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALE] <value>
```

```
DISPlay:TXGain:Y[:SCALE]?
```

## Arguments

<value>::=<NRf> specifies the vertical scale in dB.

## Returns

A floating point value that is the vertical scale in dB.

## Examples

```
DISPLAY:TXGAIN:Y:SCALE 100 sets the vertical scale to 100 dB.
```

## DISPlay:TXGain:Y[:SCALE]:AUTO (No Query Form)

Automatically scales the plot vertically.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

```
DISPLAY:TXGAIN:Y:SCALE:AUTO automatically scales the plot vertically.
```

## DISPlay:TXGain:Y[:SCALE]:BOTTom

Sets or queries the bottom of the vertical position of the display.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALE]:BOTTom <value>
```

```
DISPlay:TXGain:Y[:SCALE]:BOTTom?
```

## Arguments

<value>::=<NRf> specifies the bottom of the vertical position of the display.

## Returns

A floating point value that is the bottom of the vertical position of the display.

## Examples

```
DISPLAY:TXGAIN:Y:SCALE:BOTTOM -100 sets the bottom of Y axis to -100 dB.
```

## DISPlay:TXGain:Y[:SCALE]:PDIVision

Sets or queries the vertical scale (per division) of the graph.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALE]:PDIVision <value>
```

```
DISPlay:TXGain:Y[:SCALE]:PDIVision?
```

## Arguments

<value>::=<NRf> specifies the vertical scale (per division).

## Returns

A floating point value that is the vertical scale (per division) of the graph.

## Examples

```
DISPLAY:TXGAIN:Y:SCALE:PDIVision 0.5 sets the vertical scale to 0.5 dB/div.
```

## DISPlay:TXGain:Y[:SCALE]:POSITION

Sets or queries the vertical position of the trace.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALe]:POSition <value>
```

```
DISPlay:TXGain:Y[:SCALe]:POSition?
```

## Arguments

<value> ::= <NRf> specifies the vertical position.

## Returns

A floating point value that is the vertical position.

## Examples

DISPLAY:TXGAIN:Y:SCALE:PDIVision 0 sets the vertical position to 0 dB.

## DISPlay:TXGain:Y[:SCALe]:TOP

Sets or queries the top of the vertical scale.

## Conditions

Measurement view: Transmission Gain

## Group

Display commands

## Syntax

```
DISPlay:TXGain:Y[:SCALe]:TOP <value>
```

```
DISPlay:TXGain:Y[:SCALe]:TOP?
```

## Arguments

<value> ::= <NRf> specifies the top of the vertical scale.

## Returns

A floating point value that is the top of the vertical scale.

## Examples

DISPLAY:TXGAIN:Y:SCALE:TOP 10 sets the vertical top to 10 dB.



## **:DISPlay:WINDow:ACQuisition:BANdwidth:SHOw:STAtE {set and query}**

Sets or queries the state of show acquisition bandwidth checkbox.

### **Conditions**

Measurement views: Spectrum, Spectrogram

### **Group**

Display commands

### **Syntax**

```
:DISPlay:WINDow:ACQuisition:BANdwidth:SHOw:STAtE {0|1|TRUE |FALSE}
```

```
:DISPlay:WINDow:ACQuisition:BANdwidth:SHOw:STAtE?
```

### **Argument**

0 is used to change 'Show Acquisition Bandwidth' checkbox state from checked to unchecked.

1 is used to change 'Show Acquisition Bandwidth' checkbox state from unchecked to checked.

### **Examples**

:DISPlay:WINDow:ACQuisition:BANdwidth:SHOw:STAtE 1 changes the state of 'Show Acquisition Bandwidth' checkox to checked.

## **DISPlay:WINDow:ACTive:MEASurement? (Query Only)**

Queries the active measurement views.

### **Conditions**

Measurement views: All

### **Group**

Display commands

### **Syntax**

```
DISPlay:WINDow:ACTive:MEASurement?
```

### **Arguments**

None

### **Returns**

```
<view1>,<view2>,...,<view(n)>
```

Where

<view(n)>::<string> is the view name as shown in the following table.

**Table 40: Measurement view mnemonic**

Return value	Measurement view	Display group
"SPEC	Spectrum	General signal viewing
"DPX"	DPX (Digital Phosphor) spectrum	
"MAGVT"	Amplitude versus Time	
"ACP"	Channel power and ACPR	
"FVT"	Frequency versus Time	
"PHVT"	Phase versus Time	
"IQVT"	RF I&Q versus Time	
"SGRam"	Spectrogram	
"TOV"	Time overview	
"AM"	Amplitude modulation	
"FM"	Frequency modulation	
"PM"	Phase modulation	
"CONS"	Constellation	General purpose digital modulation
"DIQV"	Demodulated I&Q versus Time	
"EDI"	Eye Diagram	
"EVM"	EVM versus Time	
"FVDT"	Frequency deviation versus Time	
"MERR"	Magnitude error versus Time	
"PERR"	Phase error versus Time	
"SIGN"	Signal quality	
"STAB"	Symbol table	
"TDI"	Trellis Diagram	
"CCDF"	CCDF	RF measurements
"MCP"	MCPR (Multiple Carrier Power Ratio)	
"OBW"	Occupied bandwidth	
"PNO"	Phase noise	
"SPUR"	Spurious	
"STAT"	Pulse statistics	Pulsed RF
"RES"	Pulse table (results table)	
"TRAC"	Pulse trace	

**DISPlay: WINDow: APPLy: SOURce: COLOr {set and query}**

Sets or queries the state of apply source color to trace checkbox.

**Conditions**

Measurement views: All

Applicable for only MSO/LPD/SX.

## Group

Display commands

## Syntax

```
:DISPlay:WINDow:APPLy:SOURce:COLOr {0|1|TRUE |FALSE}
```

```
:DISPlay:WINDow:APPLy:SOURce:COLOr?
```

## Argument

0 is used to change 'Apply Source Color to Trace' checkbox state from unchecked to checked.

1 is used to change 'Apply Source Color to Trace' checkbox state from checked to unchecked.

## Examples

```
:DISPlay:WINDow:APPLy:SOURce:COLOr 1 changes the state of 'Apply Source Color to Trace' checkox to checked.
```

## DISPlay:WINDow:COLor:SCHeme

Sets or queries the color scheme for displaying traces and background on the screen.

## Conditions

Measurement views: All

## Group

Display commands

## Syntax

```
DISPlay:WINDow:COLor:SCHeme { THUNderstorm | BLIZzard | CLASsic }
```

```
DISPlay:WINDow:COLor:SCHeme?
```

## Arguments

THUNderstorm displays the background in dark blue.

BLIZzard displays the background in white. It saves ink when printing the screen image.

CLASsic displays the background in black (default).

## Examples

```
DISPLAY:WINDOW:COLOR:SCHEME BLIZzard displays the background in white.
```

## DISPlay:WINDow:OPTimized:MEASurement? (Query Only)

Queries the measurement views that are optimized. "Optimized" means that there is a perfect match between the view's settings and the actual acquisition parameters to meet the specifications. When multiple measurements are running at one time, the measurements can

have different requirements for setting the acquisition hardware. You can make a measurement optimized by selecting it using the following commands:

- *DISPlay:GENeral:MEASview:SElect* for the general signal viewing
- *DISPlay:DDEMod:MEASview:SElect* for the digital modulation views
- *DISPlay:GPRF:MEASview:SElect* for the RF measurement views
- *DISPlay:PULSe:MEASview:SElect* for the pulsed RF measurement views

## Conditions

Measurement views: All

## Group

Display commands

## Syntax

`DISPlay:WINDow:OPTimized:MEASurement?`

## Arguments

None

## Returns

`<view1>, <view2>, ..., <view(n)>`

Where

`<view(n)>::<string>` is the view name as shown in the table. (See Table 2-39 on page 2-812.)

## Examples

`DISPLAY:WINDOW:OPTIMIZED:MEASUREMENT?` might return "SPEC", "MCP", indicating that the views of spectrum and MCPR are optimized.

## **DISPlay:WLAN:AD:CONSte:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticules in the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Display commands

## Syntax

`DISPlay:WLAN:AD:CONSte:WINDow:TRACe:GRATicule:GRID:STATe { 0 | 1 | ON | OFF }`

`DISPlay:WLAN:AD:CONSte:WINDow:TRACe:GRATicule:GRID:STATe?`

## Arguments

0 or OFF hides the graticules in the graph

1 or ON shows the graticules in the graph

## Returns

0 or OFF means that the graticules in the graph are hidden

1 or ON means that the graticules in the graph are showing

## Examples

`DISPLAY:WLAN:AD:CONSTE:WINDOW:TRACE:GRATICULE:GRID:STATE 1` shows the graticules in the display.

## DISPlay:WLAN:AD:CONSte:WINDow:TRACe:MARKer:STATe

Determines whether to show or hide marker readout in graph in the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AD:CONSte:WINDow:TRACe:MARKer:STATe { 0 | 1 | ON | OFF }
```

```
DISPlay:WLAN:AD:CONSte:WINDow:TRACe:MARKer:STATe?
```

## Arguments

OFF or 0 hides the marker readout in the graph

ON or 1 shows the marker readout in the graph

## Returns

0 or OFF means that the graticules in the graph are hidden

1 or ON means that the graticules in the graph are shown

## Examples

`DISPLAY:WLAN:AD:CONSTE:WINDOW:TRACE:MARKER:STATE 1` shows the marker readout in the display.

## DISPlay:WLAN:AD:CONSte:ZOOM:SCALE

Sets or queries the zoom scale value in the Scale tab of the Settings control panel for the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Display commands

## Syntax

DISPlay:WLAN:AD:CONStE:ZOOM:SCALe <value>

DISPlay:WLAN:AD:CONStE:ZOOM:SCALe?

## Arguments

<value>=<NRf> where range is 0.1 to 10

## Returns

<NRf> where range is 0.1 to 10

## Examples

DISPLAY:WLAN:AD:CONSTE:ZOOM:SCALE 5 sets the zoom scale value to 5.

## DISPlay:WLAN:AD:CONStE:ZOOM[:SCALe]:X:OFFSet

Sets or queries the Horizontal Position value in the Scale tab of the Settings control panel for the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Display commands

## Syntax

DISPlay:WLAN:AD:CONStE:ZOOM[:SCALe]:X:OFFSet <value>

DISPlay:WLAN:AD:CONStE:ZOOM[:SCALe]:X:OFFSet?

## Arguments

<value>=<NRf> where range is -5.1 to 5.1

## Returns

<NRf> where range is -5.1 to 5.1

## Examples

DISPLAY:WLAN:AD:CONSTE:ZOOM[:SCALE]:X:OFFSET 3.5 sets the Horizontal Position to 3.5.

## DISPlay:WLAN:AD:CONStE:ZOOM[:SCALe]:Y:OFFSet

Sets or queries the Vertical Position value in the Scale tab of the Settings control panel for the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AD:CONStE:ZOOM[:SCALE]:Y:OFFSet <value>
```

```
DISPlay:WLAN:AD:CONStE:ZOOM[:SCALE]:Y:OFFSet?
```

## Arguments

<value>=<NRf> where range is -5.1 to 5.1

## Returns

<NRf> where range is -5.1 to 5.1

## Examples

DISPLAY:WLAN:AD:CONSTE:ZOOM[:SCALE]:Y:OFFSET-3.5 sets the Vertical Position to -3.5.

## DISPlay:WLAN:AD:DATA:RADix

Set or queries the symbol radix type (set in the Prefs tab of the Settings control panel) in the display.

## Conditions

Measurement views: All 802.11ad Symbol Table and Constellation

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AD:DATA:RADix {DEC,BIN}
```

```
DISPlay:WLAN:AD:DATA:RADix?
```

## Arguments

DEC sets the radix type to Decimal

BIN sets the radix type to Binary

## Returns

See Arguments

## Examples

DISPLAY:WLAN:AD:DATA:RADIX might return BIN, indicating the data radix selected is binary.

## **DISPlay:WLAN:AD:EVM:SCALE:AUtO (No Query Form)**

Rescales both horizontal and vertical axis automatically to fit the waveform to the screen in the display.

### **Conditions**

Measurement views: 802.11ad EVM vs Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:AD:EVM:SCALE:AUtO
```

### **Arguments**

None

### **Returns**

None

### **Examples**

DISPLAY:WLAN:AD:EVM:SCALE:AUtO fits the waveform to the graph both horizontally and vertically.

## **DISPlay:WLAN:AD:EVM:WINDow:TRACe:GRATicule:GRID:STATe**

Determines whether to show or hide the graticule in the display.

### **Conditions**

Measurement views: 802.11ad EVM vs Time

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:AD:EVM:WINDow:TRACe:GRATicule:GRID:STATe { 0 | 1 | ON | OFF }
```

```
DISPlay:WLAN:AD:EVM:WINDow:TRACe:GRATicule:GRID:STATe ?
```

### **Arguments**

OFF or 0 hides the graticules in the graph

1 or ON shows the graticules in the graph

### **Returns**

0 or OFF means that the graticules in the graph are hidden

1 or ON means that the graticules in the graph are showing



## Examples

```
DISPLAY:WLAN:AD:EVM:WINDOW:TRACE:GRATICULE:GRID:STATE OFF
```

turns off the graticule in the graph.

## DISPlay:WLAN:AD:EVM:WINDow:TRACe:MARKer:STATe

Determines whether to show or hide the readout for the selected marker in the display.

### Conditions

Measurement views: 802.11ad EVM vs Time

### Group

Display commands

### Syntax

```
DISPlay:WLAN:AD:EVM:WINDow:TRACe:MARKer:STATe { 0 | 1 | ON | OFF }
```

```
DISPlay:WLAN:AD:EVM:WINDow:TRACe:MARKer:STATe ?
```

### Arguments

OFF or 0 hides the marker readout in the graph

1 or ON shows the marker readout in the graph

### Returns

0 or OFF means that the graticules in the graph are hidden

1 or ON means that the graticules in the graph are shown

## Examples

```
DISPLAY:WLAN:AD:EVM:WINDOW:TRACE:MARKER:STATE ON shows the readout for the selected marker in the graph.
```

## DISPlay:WLAN:AD:EVM:X:SCALE

Sets or queries the horizontal scale (time) axis range in the display.

### Conditions

Measurement views: 802.11ad EVM vs Time

### Group

Display commands

### Syntax

```
DISPlay:WLAN:AD:EVM:X:SCALE <value>
```

```
DISPlay:WLAN:AD:EVM:X:SCALE?
```

## Arguments

<value>=<NRf>

## Returns

<NRf> = time trace

## Examples

DISPLAY:WLAN:AD:EVM:X:SCALE 30E-3 sets the horizontal scale to 30 ms.

## DISPlay:WLAN:AD:EVM:X:SCALE:AUtO (No Query Form)

Rescales the horizontal axis automatically to fit the waveform to the screen in the display.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Display commands

## Syntax

DISPlay:WLAN:AD:EVM:X:SCALE:AUtO

## Arguments

None

## Returns

None

## Examples

DISPLAY:WLAN:AD:EVM:X:SCALE:AUtO rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:WLAN:AD:EVM:X:SCALE:OFFSet

Sets or queries the start of the horizontal offset value (position) of the trace in the display.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Display commands

## Syntax

DISPlay:WLAN:AD:EVM:X:SCALE:OFFSet <value>

---

DISPlay:WLAN:AD:EVM:X:SCALE:OFFSet?

### Arguments

<value>=<NRf>

### Returns

<NRf> = position of trace

### Examples

DISPLAY:WLAN:AD:EVM:X:SCALE:OFFSET 150E-3 sets the horizontal offset value of the trace to 150 ms.

## DISPlay:WLAN:AD:EVM:Y:SCALE

Sets or queries the vertical scale (time) axis range of the in display.

### Conditions

Measurement views: 802.11ad EVM vs Time

### Group

Display commands

### Syntax

DISPlay:WLAN:AD:EVM:Y:SCALE<value>

DISPlay:WLAN:AD:EVM:Y:SCALE?

### Arguments

<value>=<NRf>

### Returns

<NRf> = vertical range

### Examples

DISPLAY:WLAN:AD:EVM:Y:SCALE 100 sets the vertical range scale to 100 dB.

## DISPlay:WLAN:AD:EVM:Y:SCALE:AUTO (No Query Form)

Rescales the vertical scale automatically to fit the waveform in the display.

### Conditions

Measurement views: 802.11ad EVM vs Time

### Group

Display commands

## Syntax

DISPlay:WLAN:AD:EVM:Y:SCALE:AUTO

## Arguments

None

## Returns

None

## Examples

DISPLAY:WLAN:AD:EVM:Y:SCALE:AUTO rescales the vertical scale to fit the waveform on the screen.

## DISPlay:WLAN:AD:EVM:Y:SCALE:OFFSet

Sets or queries the start of the vertical offset scale value (position) in the display.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Display commands

## Syntax

DISPlay:WLAN:AD:EVM:Y:SCALE:OFFSet<value>

DISPlay:WLAN:AD:EVM:Y:SCALE:OFFSet?

## Arguments

<value>=<NRf>

## Returns

<NRf> = vertical offset

## Examples

DISPLAY:WLAN:AD:EVM:Y:SCALE:OFFSET 100 sets the vertical offset to 100 dB.

## DISPlay:WLAN:AD:MEASview:DELeTe (No Query Form)

Delete (Close) the corresponding 802.11ad measurement view from the analysis window.

## Conditions

Measurement views: All 802.11ad measurement views

---

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AD:MEASview:DELeTe {CONS,SUMM,EVM,STAB}
```

## Arguments

CONS selects the 802.11ad Constellation view

SUMM selects the 802.11ad Summary view

EVM selects the 802.11ad EVM vs Time view

STAB selects the 802.11ad Symbol Table view

## Returns

See Arguments

## Examples

DISPLAY:WLAN:AD:MEASVIEW:DELETE CONS will close the Constellation display.

## DISPlay:WLAN:AD:MEASview:NEW (No Query Form)

Displays a new 802.11ad measurement view in the analysis window.

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AD:MEASview:NEW {CONS,SUMM,EVM,STAB}
```

## Arguments

CONS selects the 802.11ad Constellation view

SUMM selects the 802.11ad Summary view

EVM selects the 802.11ad EVM vs Time view

STAB selects the 802.11ad Symbol Table view

## Returns

See Arguments

## Examples

`DISPLAY:WLAN:AD:MEASVIEW:NEW CONS` will display the constellation display.

## DISPlay:WLAN:AD:MEASview:SElect

Selects the corresponding 802.11 ad measurement view on the analysis window. The query command returns the currently selected view.

### Conditions

Measurement views: All 802.11ad measurement views

### Group

Display commands

### Syntax

```
DISPlay:WLAN:AD:MEASview:SElect {CONS, SUMM, EVM, STAB}
```

```
DISPlay:WLAN:AD:MEASview:SElect?
```

### Arguments

CONS selects the 802.11ad Constellation view

SUMM selects the 802.11ad Summary view

EVM selects the 802.11ad EVM vs Time view

STAB selects the 802.11ad Symbol Table view

### Returns

See Arguments

## Examples

`DISPLAY:WLAN:AD:MEASVIEW:SELECT SUMM` selects the 802.11ad Summary view.

## DISPlay:WLAN:AY:CONSte:WINDow:TRACe:GRATicule:GRID:STATe

Set or query whether to show or hide the graticule in the Constellation display.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Display commands

### Syntax

```
DISPlay:WLAN:AY:CONSte:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 0 | 1 }
```

```
DISPlay:WLAN:AY:CONSte:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule on the graph.

ON or 1 shows the graticule on the graph.

## Returns

0 indicates the graticule is hidden on the graph.

1 indicates the graticule is shown on the graph.

## Examples

DISP:WLAN:AY:CONS:WIND:TRAC:GRAT:GRID:STAT 1 enables the graticule to show.

DISP:WLAN:AY:CONS:WIND:TRAC:GRAT:GRID:STAT? may return 0, which means the graticule is disabled.

## DISPlay:WLAN:AY:CONSte:WINDow:TRACe:MARKer:STATe

Set or query whether to show or hide the marker readout on the graph in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Display commands

## Syntax

DISPlay:WLAN:AY:CONSte:WINDow:TRACe:MARKer:STATe { OFF | ON | 0 | 1 }

DISPlay:WLAN:AY:CONSte:WINDow:TRACe:MARKer:STATe?

## Arguments

OFF or 0 hides the marker readout on the graph.

ON or 1 shows the marker readout on the graph.

## Returns

OFF or 0 means the marker readout is hidden on the graph.

ON or 1 means the marker readout shows on the graph.

## Examples

DISP:WLAN:AY:CONS:WIND:TRAC:MARK:STAT 1 enables marker readout to show.

DISP:WLAN:AY:CONS:WIND:TRAC:MARK:STAT? may return 0, which means marker readout is disabled.

## DISPlay:WLAN:AY:CONSte:ZOOM:SCALE

Set or query the scale value, which is located under the scale tab.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:CONStE:ZOOM:SCALe <value>
```

```
DISPlay:WLAN:AY:CONStE:ZOOM:SCALe?
```

## Arguments

<value> ::= <NRf> specifies the scale value.

Range: 0.1 to 10

## Returns

<NRf>

## Examples

DISP:WLAN:AY:CONS:ZOOM:SCAL 5 sets the scale value to 5.

DISP:WLAN:AY:CONS:ZOOM:SCAL? may return 6, indicating that 6 is the scale value.

## DISPlay:WLAN:AY:CONStE:ZOOM:[SCALe]:X:OFFSet

Set or query the horizontal position value of the scale in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:CONStE:ZOOM:[SCALe]:X:OFFSet <value>
```

```
DISPlay:WLAN:AY:CONStE:ZOOM:[SCALe]:X:OFFSet?
```

## Arguments

<value> ::= <NRf> specifies the horizontal position value.

Range: -3.5 to 3.5

## Returns

<NRf>



## Examples

`DISP:WLAN:AY:CONS:ZOOM:SCAL:X:OFFS 3.5` sets the horizontal position value of the scale to 3.5.

`DISP:WLAN:AY:CONS:ZOOM:SCAL:X:OFFS?` may return 3.5, which is the horizontal position value of the scale.

## DISPlay:WLAN:AY:CONSte:ZOOM:[SCALe]:Y:OFFSet

Set or query the vertical position of the scale.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Display commands

### Syntax

`DISPlay:WLAN:AY:CONSte:ZOOM:[SCALe]:Y:OFFSet <value>`

`DISPlay:WLAN:AY:CONSte:ZOOM:[SCALe]:Y:OFFSet?`

### Arguments

`<value>::=<NRf>` specifies the vertical position value.

Range: -3.5 to 3.5

### Returns

`<NRf>`

## Examples

`DISP:WLAN:AY:CONS:ZOOM:Y:OFFS -3.5` sets the vertical position value of the scale to -3.5.

`DISP:WLAN:AY:CONS:ZOOM:Y:OFFS?` may return -3.5, which is the vertical position value of the scale.

## DISPlay:WLAN:AY:DATA:RADix

Set or query the type of symbol radix present in the WLAN 802.11ay Symbol Table control panel settings.

### Conditions

Measurement views: WLAN 802.11ay Symbol Table display

### Group

Display commands

### Syntax

`DISPlay:WLAN:AY:DATA:RADix { DEC | BIN }`

`DISPlay:WLAN:AY:DATA:RADix?`

## Arguments

DEC sets the data radix type to decimal.

BIN sets the data radix type to binary.

## Returns

DEC means that the data radix type is decimal.

BIN means that the data radix type is binary.

## Examples

DISP:WLAN:AY:DATA:RAD DEC sets the data radix to decimal.

DISP:WLAN:AY:DATA:RAD? may return BIN, indicating the data radix selected is binary.

## DISPlay:WLAN:AY:EVM:SCALE:AUTO (No Query Form)

Rescales both horizontal and vertical axis automatically to fit the waveform to the screen in the 802.11ay EVM display.

### Conditions

Measurement views: 802.11ay Analysis EVM

### Group

Display commands

### Syntax

DISPlay:WLAN:AY:EVM:SCALE:AUTO

### Arguments

None

### Examples

DISP:WLAN:AY:EVM:SCALE:AUTO rescales both horizontal and vertical axis to fit the waveform to the graph.

## DISPlay:WLAN:AY:EVM:WINDOW:TRACE:GRATICULE:GRID:STATE

Set or query whether to show or hide the graticule in the 802.11ay EVM display.

### Conditions

Measurement views: 802.11ay Analysis EVM

### Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:EVM:WINDow:TRACe:GRATicule:GRID:STATe { OFF | ON | 1 | 0 }
DISPlay:WLAN:AY:EVM:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

OFF or 0 hides the graticule on the graph.

ON or 1 shows the graticule on the graph.

## Returns

OFF or 0 means the graticule is hidden on the graph.

ON or 1 means that the graticule shows on the graph.

## Examples

DISP:WLAN:AY:EVM:WIND:TRAC:GRAT:GRID:STAT OFF hides the graticule on the graph.

DISP:WLAN:AY:EVM:WIND:TRAC:GRAT:GRID:STAT? may return ON, which means the graticule shows on the graph.

## DISPlay:WLAN:AY:EVM:WINDow:TRACe:MARKer:STATe

Set or query whether to show or hide the readout for the selected marker in the 802.11ay EVM display.

## Conditions

Measurement views: 802.11ay Analysis EVM

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:EVM:WINDow:TRACe:MARKer:STATe { OFF | ON | 0 | 1 }
DISPlay:WLAN:AY:EVM:WINDow:TRACe:MARKer:STATe?
```

## Arguments

OFF or 0 hides the readout for the selected marker on the graph.

ON or 1 shows the readout for the selected marker on the graph.

## Returns

OFF or 0 means the readout for the selected marker is hidden on the graph.

ON or 1 means the readout for the selected marker shows on the graph.

## Examples

DISP:WLAN:AY:EVM:WIND:TRAC:MARK:STAT ON shows the readout for the selected marker on the graph.

DISP:WLAN:AY:EVM:WIND:TRAC:MARK:STAT? may return OFF, which means the readout for the selected marker is hidden on the graph.

## DISPlay:WLAN:AY:EVM:X:SCALe

Set or query the horizontal (time) axis range in the 802.11ay EVM display.

### Conditions

Measurement views: 802.11ay Analysis EVM

### Group

Display commands

### Syntax

DISPlay:WLAN:AY:EVM:X:SCALe <value>

DISPlay:WLAN:AY:EVM:X:SCALe?

### Arguments

<value> ::= <NRf> specifies the horizontal (time) axis range.

Units: ms (milliseconds)

### Returns

<NRf>

### Examples

DISP:WLAN:AY:EVM:X:SCAL 30e-3 sets the horizontal scale to 30 ms. DISP:WLAN:AY:EVM:X:SCAL? may return 30e-3, which means that the horizontal scale is 30 ms.

## DISPlay:WLAN:AY:EVM:X:SCALe:AUTO (No Query Form)

Rescale the horizontal axis automatically to fit the waveform to the screen in the EVM display.

### Conditions

Measurement views: 802.11ay Analysis in 802.11ay EVM

### Group

Display commands

### Syntax

DISPlay:WLAN:AY:EVM:X:SCALe:AUTO

### Arguments

None

## Examples

`DISP:WLAN:AY:EVM:X:SCALE:AUTO` rescales the horizontal scale automatically to fit the waveform to the screen.

## DISPlay:WLAN:AY:EVM:X:SCALE:OFFSet

Set or query the start of the horizontal (time) trace in the 802.11ay EVM display.

### Conditions

Measurement views: 802.11ay Analysis EVM

### Group

Display commands

### Syntax

`DISPlay:WLAN:AY:EVM:X:SCALE:OFFSet <value>`

`DISPlay:WLAN:AY:EVM:X:SCALE:OFFSet?`

### Arguments

`<value>::=<NRf>` specifies the minimum horizontal value.

Units: ms (milliseconds)

### Returns

`<NRf>`

## Examples

`DISP:WLAN:AY:EVM:X:SCALE:OFFS 150e-3` sets the horizontal offset value to 150 ms.

`DISP:WLAN:AY:EVM:X:SCALE:OFFS?` may return `150e-3`, which means the horizontal offset value is 150 ms.

## DISPlay:WLAN:AY:EVM:Y:SCALE

Set or query the vertical range in the 802.11ay EVM display.

### Conditions

Measurement views: 802.11ay Analysis EVM

### Group

Display commands

### Syntax

`DISPlay:WLAN:AY:EVM:Y:SCALE <value>`

`DISPlay:WLAN:AY:EVM:Y:SCALE?`

## Arguments

<value> ::= <NRf> specifies the vertical scale. Units: % or dB (based on the EVM units set)

## Returns

<NRf>

## Examples

DISP:WLAN:AY:EVM:Y:SCALE 100 sets the vertical range to 100 dB.

DISP:WLAN:AY:EVM:Y:SCALE? may return 100, which means that the vertical range is 100%.

## DISPlay:WLAN:AY:EVM:Y:SCALE:AUtO (No Query Form)

Rescale the vertical scale automatically to fit the waveform in the 802.11ay EVM display.

## Conditions

Measurement views: 802.11ay Analysis EVM

## Group

Display commands

## Syntax

DISPlay:WLAN:AY:EVM:Y:SCALE:AUtO

## Arguments

None

## Examples

DISP:WLAN:AY:EVM:X:SCALE:AUtO rescales the vertical scale automatically to fit the waveform to the screen.

## DISPlay:WLAN:AY:EVM:Y:SCALE:OFFSet

Set or query the start of the vertical (EVM) trace in the 802.11ay EVM display. This command corresponds to setting or query positions in the EVM display.

## Conditions

Measurement views: 802.11ay Analysis EVM

## Group

Display commands

## Syntax

DISPlay:WLAN:AY:EVM:Y:SCALE:OFFSet <value>

DISPlay:WLAN:AY:EVM:Y:SCALE:OFFSet?

## Arguments

<value>::=<NRf> specifies the vertical range.

Units: % or dB (based on the EVM units set)

## Returns

<NRf>

## Examples

DISP:WLAN:AY:EVM:Y:SCAL:OFFS 0 sets the vertical offset to 0%.

DISP:WLAN:AY:EVM:Y:SCAL:OFFS? may return 0, which means the vertical offset is 0 dB.

## DISPlay:WLAN:AY:MEASview:DELeTe (No Query Form)

Deletes the corresponding WLAN 802.11ay display from the analysis window.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Display commands

## Syntax

DISPlay:WLAN:AY:MEASview:DELeTe { CONS | SUMM | EVM | STAB }

## Arguments

Select one of the below arguments:

Arguments	Description
CONS	Selects the WLAN 802.11ay Constellation view.
SUMM	Selects the WLAN 802.11ay Summary view.
EVM	Selects the WLAN 802.11ay EVM versus Time view.
STAB	Selects the WLAN 802.11ay Symbol Table view.

## Examples

DISP:WLAN:AY:MEAS:DEL CONS deletes the Constellation display.

## DISPlay:WLAN:AY:MEASview:NEW (No Query Form)

Shows a new WLAN 802.11ay display on the analysis window.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:MEASview:NEW { CONS | SUMM | EVM | STAB }
```

## Arguments

Select one of the below arguments:

Argument	Description
CONS	Selects the WLAN 802.11ay Constellation view.
SUMM	Selects the WLAN 802.11ay Summary view.
EVM	Selects the WLAN 802.11ay EVM versus Time view.
STAB	Selects the WLAN 802.11ay Symbol Table view.

## Examples

DISP:WLAN:AY:MEAS:NEW CONS opens the Constellation display.

## DISPlay:WLAN:AY:MEASview:SElect

Set or query the corresponding 802.11 ay display on the analysis window. The query command returns the currently selected view.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Display commands

## Syntax

```
DISPlay:WLAN:AY:MEASview:SElect { CONS | SUMM | EVM | STAB }
```

```
DISPlay:WLAN:AY:MEASview:SElect?
```

## Arguments

Select one of the below arguments:

## Arguments

Select one of the below arguments:

Arguments	Description
CONS	Selects the WLAN 802.11ay Constellation view.
SUMM	Selects the WLAN 802.11ay Summary view.
EVM	Selects the WLAN 802.11ay EVM versus Time view.
STAB	Selects the WLAN 802.11ay Symbol Table view.



## Returns

See Arguments.

## Examples

`DISP:WLAN:AY:MEAS:SEL SUMM` selects the Summary view.

`DISP:WLAN:AY:MEAS:SEL?` may return STAB, which indicates the Symbol Table view.

## DISPlay:WLAN:CONSte[:SCALe]

Sets the magnification value (zoom factor) for the WLAN constellation plot.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:CONSte[:SCALe] <NRf>`

`DISPlay:WLAN:CONSte[:SCALe]?`

## Arguments

Floating point number that represents the scale factor. Valid inputs are any floating point number between 0.1 and 10. A positive value zooms in and a negative value zooms out.

## Returns

A floating point number indicating the current magnification value.

## Examples

`DISP:WLAN:CONS:SCAL 5.0` zooms the display out by a factor of 5.

## DISPlay:WLAN:CONSte:X:OFFSet

Sets or queries the value of the offset of the horizontal axis from the center of the WLAN Constellation display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:CONStE:X:OFFSet <NRf>
```

```
DISPlay:WLAN:CONStE:X:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the Constellation plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 recenters the plot.

## Returns

The value of the offset of the X axis from the center for the Constellation plot as a floating point number.

## Examples

```
DISP:WLAN:CONST:X:OFFS 1 shifts the constellation display to the left by 1.
```

## DISPlay:WLAN:CONStE:Y:OFFSet

Sets or queries the value of the offset of the vertical axis from the center for the WLAN Constellation display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:CONStE:Y:OFFSet <NRf>
```

```
DISPlay:WLAN:CONStE:Y:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the constellation plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

## Returns

The value of the offset of the Y axis from the center for the Constellation display as a floating point number.

## Examples

```
DISP:WLAN:CONST:Y:OFFS 1 shifts the constellation display up by 1.
```

## DISPlay:WLAN:CRESpOse:AUTO (No Query Form)

Automatically rescales the horizontal and vertical values for the best display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:AUTO
```

### Arguments

None

### Examples

DISP:WLAN:CRESP:AUTO automatically rescales the horizontal and vertical values.

## DISPlay:WLAN:CRESpOse:FREQuency:AUTO (No Query Form)

Automatically rescales the horizontal subcarrier (frequency) start and stop values to fit the waveform to the screen in the WLAN Channel Response display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:FREQuency:AUTO
```

### Arguments

None

### Examples

DISP:WLAN:CRESP:FREQ:AUTO automatically rescales the horizontal subcarrier (frequency) start and stop values to fit the waveform to the screen.

## **DISPlay:WLAN:CRESpOse:FREQuency:STARt**

Specifies or queries the horizontal start value (left edge) on the WLAN Channel Response display. The units are either Frequency (Hz) or Subcarriers, and are set using the command [SENSe]:WLAN:UNIT:FREQuency.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:CRESpOse:FREQuency:STARt <NRf>
```

```
DISPlay:WLAN:CRESpOse:FREQuency:STARt?
```

### **Arguments**

Floating point number that specifies the horizontal start value (left edge) on the graph.

### **Returns**

The returned value indicates the Subcarrier (for non-b standards only) or frequency shown at the left edge of the display.

### **Examples**

`DISP:WLAN:CRES:FREQ:START 2.41e9` sets the horizontal start value (left edge) to 2.41 GHz (assuming the units have been set to Frequency).

## **DISPlay:WLAN:CRESpOse:FREQuency:STOP**

Specifies or queries the horizontal stop value (right edge) on the WLAN Channel Response display. The units are either Frequency (Hz) or Subcarriers, and are set using the command [SENSe]:WLAN:UNIT:FREQuency.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:CRESpOse:FREQuency:STOP <NRf>
```

```
DISPlay:WLAN:CRESpOse:FREQuency:STOP?
```

### **Arguments**

Floating point number that specifies the horizontal stop value (right edge) on the display.

## Examples

`DISP:WLAN:CRES:FREQ:STOP 8.1250e9` sets the horizontal stop value to 8.1250 GHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:CRESpOse:MAGNitude:AUTO (No Query Form)

Rescales the vertical magnitude offset (position) and scale values to automatically to fit the waveform to the screen in the WLAN Channel Response magnitude graph.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:MAGNitude:AUTO
```

### Arguments

None

## Examples

`DISP:WLAN:CRES:MAGN:AUTO` rescales the vertical magnitude offset (position) and scale values automatically to fit the waveform to the screen.

## DISPlay:WLAN:CRESpOse:MAGNitude:OFFSet

Sets or queries the vertical offset (position in the center of the vertical axis) in the WLAN Channel Response magnitude graph.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:MAGNitude:OFFSet <NRf>
```

```
DISPlay:WLAN:CRESpOse:MAGNitude:OFFSet?
```

### Arguments

Floating point number that specifies the vertical offset in dB.

## Examples

`DISP:WLAN:CRES:MAGN:OFFS 5` sets the vertical offset to 5 dB.

## **DISPlay:WLAN:CRESpOse:MAGNitude[:SCALe]**

Sets or queries the value of the vertical scale for the WLAN Channel Response magnitude display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:CRESpOse:MAGNitude[:SCALe] <Nrf>`

`DISPlay:WLAN:CRESpOse:MAGNitude[:SCALe]?`

### Arguments

Floating point number that represents the vertical scale value in dB.

## Examples

`DISP:WLAN:CRES:MAGN:SCAL 5.0` sets the vertical scale to 5 dB.

## **DISPlay:WLAN:CRESpOse:MARKer:SHOW:STATe**

Shows or hides the marker readout on the WLAN Channel Response display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:CRESpOse:MARKer:SHOW:STATe {0|1|OFF|ON}`

`DISPlay:WLAN:CRESpOse:MARKer:SHOW:STATe?`

### Arguments

ON or 1 specifies to show the marker readout.

OFF or 0 specifies to turn the marker readout off.

## Examples

`DISP:WLAN:CRES:MARK:SHOW:STATE ON` specifies to turn the marker readout on.

## DISPlay:WLAN:CRESpOse:PHASe:AUTO (No Query Form)

Rescales the vertical phase value automatically to fit the waveform to the screen in the WLAN Channel Response display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:PHASe:AUTO
```

### Arguments

None

## Examples

`DISP:WLAN:CRES:PHAS:AUTO` rescales the vertical phase value automatically to fit the waveform to the screen.

## DISPlay:WLAN:CRESpOse:PHASe:OFFSet

Sets or queries the vertical offset (position in the center of the vertical axis) in the WLAN Channel Response phase graph.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:CRESpOse:PHASe:OFFSet <NRf>
```

```
DISPlay:WLAN:CRESpOse:PHASe:OFFSet?
```

### Arguments

Floating point number that specifies the vertical offset (position) in degrees.

## Examples

`DISPLAY:WLAN:CRESPONSE:PHASE:OFFSET -14.5` sets the vertical offset value to  $-14.5^\circ$  in the WLAN Channel Response phase graph.

## **DISPlay:WLAN:CRESponse:PHASe[:SCALE]**

Sets or queries the phase value for the WLAN Channel Response display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:CRESponse:PHASe[:SCALE] <NRf>`

`DISPlay:WLAN:CRESponse:PHASe[:SCALE]?`

### Arguments

Floating point number that represents the phase value in degrees.

## Examples

`DISP:WLAN:CRES:PHASE:SCAL 5` sets the vertical scale to 5 degrees for the display.

## **DISPlay:WLAN:CRESponse:WINDow:SElect:PLOT**

Specifies or queries which graphs are displayed in the WLAN Channel Response display view. Select from Magnitude, Phase, or Both.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:CRESponse:WINDow:SElect:PLOT {MAGNitude|PHASe|BOTH}`

`DISPlay:WLAN:CRESponse:WINDow:SElect:PLOT?`

### Arguments

`MAGNitude` displays the Magnitude graph.



PHASe displays the Phase graph.

BOTH displays both graphs.

## Examples

DISP:WLAN:CRES:WIND:SEL:PLOT PHAS displays the Phase graph on the WLAN Channel Response analysis display.

## DISPlay:WLAN:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe

Sets or queries whether to show the graticule grid on the WLAN Channel Response analysis display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

DISPlay:WLAN:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe {0|1|OFF|ON}

DISPlay:WLAN:CRESpOse:WINDow:TRACe:GRATicule:GRID:STATe?

### Arguments

ON or 1 specifies to show the graticule.

OFF or 0 specifies to turn it off.

### Examples

SENSE:WLAN:CRES:WIND:TRAC:GRAT:GRID:STATE ON specifies to turn the graticule on.

## DISPlay:WLAN:EVM:FREQuency:AUTO (No Query Form)

Automatically rescales the horizontal subcarrier (frequency) start and stop values to fit the waveform to the screen in the WLAN EVM display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

DISPlay:WLAN:EVM:FREQuency:AUTO

## Arguments

None

## Examples

`DISP:WLAN:EVM:FREQ:AUTO` automatically rescales the horizontal Subcarrier (frequency) values to fit the waveform to the screen.

## DISPlay:WLAN:EVM:FREQuency:RESet (No Query Form)

Automatically resets the horizontal subcarrier (frequency) start and stop values for the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:FREQuency:RESet
```

## Arguments

None

## Examples

`DISP:WLAN:EVM:FREQ:RES` automatically resets the horizontal subcarrier start and stop values for the WLAN EVM display.

## DISPlay:WLAN:EVM:FREQuency:STARt

Specifies or queries the horizontal start value (left edge) on the WLAN EVM display. The units are either Frequency (Hz) or Subcarriers, and are set using the command `[SENSe]:WLAN:UNIT:FREQuency`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:FREQuency:STARt <NRf>
```

```
DISPlay:WLAN:EVM:FREQuency:STARt?
```

## Arguments

Floating point number that specifies the horizontal start value (left edge) on the graph.

## Returns

The returned value indicates the Subcarrier (for non-b standards only) or frequency shown at the left edge of the display.

## Examples

`DISP:WLAN:EVM:FREQ:START 2.41e9` sets the horizontal start value (left edge) to 2.41e9 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:EVM:FREQuency:STOP

Specifies or queries the horizontal stop value (right edge) on the WLAN EVM display. The units are either Frequency (Hz) or Subcarriers, and are set using the command `[SENSe]:WLAN:UNIT:FREQuency`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:EVM:FREQuency:STOP <Nrf>`

`DISPlay:WLAN:EVM:FREQuency:STOP?`

## Arguments

Floating point number that specifies the horizontal stop value (right edge) on the display.

## Examples

`DISP:WLAN:EVM:FREQ:STOP 8.1250` sets the horizontal stop value to 8.1250 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:EVM:MARKer:SHOW:STATe

Shows or hides the Marker readout on the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:MARKer:SHOW:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:EVM:MARKer:SHOW:STATe?
```

## Arguments

ON or 1 specifies to show the Marker readout.

OFF or 0 specifies to turn the Marker readout off.

## Examples

```
DISP:WLAN:EVM:MARK:SHOW:STATE ON specifies to turn the Marker readout on.
```

## DISPlay:WLAN:EVM[:SCALE]:AUTO (No Query Form)

Automatically adjusts all vertical and horizontal values to provide the best display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM[:SCALE]:AUTO
```

## Arguments

None

## Examples

```
DISP:WLAN:EVM:SCALE:AUTO automatically adjusts all vertical and horizontal values to provide the best display.
```

## DISPlay:WLAN:EVM:TIME:AUTO (No Query Form)

Sets the horizontal start and stop values to fit the waveform to the WLAN EVM display, and also enables Auto horizontal scaling so the waveform will remain properly scaled as burst lengths change.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:TIME:AUTO
```

## Arguments

None

## Examples

`DISP:WLAN:EVM:TIME:AUTO` sets the horizontal start and stop values to fit the waveform to the display, and also enables Auto horizontal scaling.

## DISPlay:WLAN:EVM:TIME:AUTO:STATe

When enabled, sets the horizontal symbol scale value automatically for the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:TIME:AUTO:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:EVM:TIME:AUTO:STATe?
```

## Arguments

`ON` or `1` sets the horizontal symbol scale value automatically.

`OFF` or `0` specifies to stop setting the horizontal symbol scale value automatically.

## Examples

`DISP:WLAN:EVM:TIME:AUTO:STATE ON` specifies to the horizontal symbol scale value automatically.

## DISPlay:WLAN:EVM:TIME:RESet (No Query Form)

Resets the horizontal symbol start and stop values for the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:EVM:TIME:RESet

## Arguments

None

## Examples

DISP:WLAN:EVM:TIME:RES resets the horizontal symbol start and stop values for the WLAN EVM display.

## DISPlay:WLAN:EVM:TIME:START

Sets or queries the horizontal symbol start value for the WLAN EVM display. The units are either Symbols or Seconds, and are set using the command [SENSe]:WLAN:UNIT:TIME.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:EVM:TIME:START <Nrf>

DISPlay:WLAN:EVM:TIME:START?

## Arguments

Floating point number.

## Examples

DISP:WLAN:EVM:TIME:START 40 sets the horizontal symbol start value of the WLAN EVM display to 40 Symbols (assuming the units have been set to Symbols).

## DISPlay:WLAN:EVM:TIME:STOP

Sets or queries the horizontal symbol stop value for the WLAN EVM display. The units are either Symbols or Seconds, and are set using the command [SENSe]:WLAN:UNIT:TIME.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:TIME:STOP <Nrf>
```

```
DISPlay:WLAN:EVM:TIME:STOP?
```

## Arguments

Floating point number.

## Examples

`DISP:WLAN:EVM:TIME:STOP 804.96e-6` sets the horizontal symbol stop value for the WLAN EVM display to 804.960  $\mu$ s (assuming the units have been set to Seconds).

## DISPlay:WLAN:EVM:WINDow:SElect:PLOT

Specifies or queries which graphs are displayed in the WLAN EVM display. Select from Symbols, Subcarriers, or Both.

To set the Time units, use `[SENSe]:WLAN:UNIT:TIME`. To set the Frequency units, use `[SENSe]:WLAN:UNIT:FREQUENCY`. To set the Radix, use `[SENSe]:WLAN:RADix`.

### **NOTE.**

*Use this command only with multicarrier standards and not singlecarrier (802.11b).*

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:WINDow:SElect:PLOT {SYMBOLs|SCARier|BOTH}
```

```
DISPlay:WLAN:EVM:WINDow:SElect:PLOT?
```

## Arguments

`SYMBOLs` displays the Symbols graph.

`SCARier` displays the Subcarriers graph.

`BOTH` displays both graphs.

## Examples

`DISP:WLAN:EVM:WIND:SEL:PLOT SCAR` displays the Subcarrier graph on the WLAN EVM display.

## DISPlay:WLAN:EVM:WINDow:TRACe:GRATicule:GRID:STATe

Shows or hides the graticule grid on the screen.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:WINDow:TRACe:GRATicule:GRID:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:EVM:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

ON or 1 specifies to show the graticule.

OFF or 0 specifies to turn it off.

## Examples

```
DISP:WLAN:EVM:WIND:TRAC:GRAT:GRID:STATE ON specifies to turn the graticule on.
```

## DISPlay:WLAN:EVM:Y:AUTO (No Query Form)

Automatically sets the vertical scale and position values for the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:Y:AUTO
```

## Arguments

None

## Examples

```
DISP:WLAN:EVM:Y:AUTO automatically sets the vertical scale and position values for the WLAN EVM display.
```

## DISPlay:WLAN:EVM:Y:OFFSet

Sets or queries the value of the offset from the center (vertical position), in percent terms, for the WLAN EVM display. To set the vertical scale, use the command `DISPlay:WLAN:EVM:Y[:SCALE]`.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:Y:OFFSet <NRf>
```

```
DISPlay:WLAN:EVM:Y:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the EVM plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

## Returns

The value of the offset from the center of the Y axis for the Error Magnitude Vector plot as a floating point number.

## Examples

```
DISP:WLAN:EVM:Y:OFFS 150 sets the offset from the center (vertical position) of the EVM plot to 150 %.
```

## DISPlay:WLAN:EVM:Y:RESet (No Query Form)

Resets the vertical scale and position (offset) values for the WLAN EVM display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:EVM:Y:RESet
```

## Arguments

None

## Examples

```
DISP:WLAN:EVM:Y:RES specifies to reset the vertical scale and position (offset) values for the WLAN EVM display.
```

## DISPlay:WLAN:EVM:Y[:SCALE]

Sets or queries the vertical scale value for the WLAN EVM display. To set the vertical position (offset) value, use the command [DISPlay:WLAN:EVM:Y:OFFSet](#).

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:EVM:Y[:SCALE] <Nrf>
```

```
DISPlay:WLAN:EVM:Y[:SCALE]?
```

### Arguments

Floating point number that specifies the vertical scale value for the WLAN EVM display.

### Examples

```
DISP:WLAN:EVM:Y:SCALE 10
```

 sets the vertical scale value for the WLAN EVM display to 10.

## DISPlay:WLAN:FLATness:AUTO (No Query Form)

Automatically rescales the horizontal and vertical axes for the best display in the WLAN Spectral Flatness view.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:FLATness:AUTO
```

### Arguments

None

### Examples

```
DISP:WLAN:FLAT:AUTO
```

 automatically rescales the horizontal and vertical axes for the best display.

## DISPlay:WLAN:FLATness:X:AUTO (No Query Form)

Rescales the horizontal axis automatically in the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:FLATness:X:AUTO
```

### Arguments

None

### Examples

DISP:WLAN:FLAT:X:AUTO rescales the horizontal axis automatically for the best display.

## DISPlay:WLAN:FLATness:X:OFFSet

Sets or queries the value of the offset of the horizontal axis from the center of the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:FLATness:X:OFFSet <NRf>
```

```
DISPlay:WLAN:FLATness:X:OFFSet?
```

### Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 recenters the plot.

## Returns

The value of the offset of the X axis from the center of the plot as a floating point number.

## Examples

`DISP:WLAN:FLAT:X:OFFS 150` shifts the display to the left by 150 %.

## DISPlay:WLAN:FLATness:X[:SCALE]

Sets or queries the horizontal scale value in the WLAN Spectral Flatness display. The units are Frequency or Subcarrier. To set the units, use `[SENSe]:WLAN:UNIT:FREQuency`



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:FLATness:X[:SCALE] <NRf>`

`DISPlay:WLAN:FLATness:X[:SCALE]?`

## Arguments

Floating point number that represents the horizontal scale.

## Examples

`DISP:WLAN:FLAT:X:SCALE 32` sets the horizontal scale value to 32.

## DISPlay:WLAN:FLATness:Y:AUTO (No Query Form)

Automatically rescales the vertical axis scale and position values in the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:FLATness:Y:AUTO
```

## Arguments

None

## Examples

DISP:WLAN:FLAT:Y:AUTO automatically selects the vertical scale and position values.

## DISPlay:WLAN:FLATness:Y:OFFSet

Sets or queries the value of the offset from the center (vertical position), in percent terms, for the WLAN Spectral Flatness display. To set the Vertical Scale, use the command [DISPlay:WLAN:FLATness:Y\[:SCALE\]](#).



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:FLATness:Y:OFFSet <NRf>
```

```
DISPlay:WLAN:FLATness:Y:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

## Returns

The value of the offset from the center of the Y axis for the plot as a floating point number.

## Examples

DISP:WLAN:FLAT:Y:OFFS 150 sets the offset from the center (vertical position) of the plot to 150 %.

## DISPlay:WLAN:FLATness:Y[:SCALE]

Sets or queries the vertical scale value for the WLAN Spectral Flatness display. To set the vertical position (offset) value, use the command [DISPlay:WLAN:FLATness:Y:OFFSet](#).



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:FLATness:Y[:SCALE] <NRf>
```

```
DISPlay:WLAN:FLATness:Y[:SCALE]?
```

### Arguments

Floating point number that specifies the vertical scale value.

### Examples

DISP:WLAN:FLAT:Y:SCALE 10 sets the vertical scale value to 10.

## DISPlay:WLAN:MEASview:DElete (No Query Form)

Deletes the selected measurement view in the WLAN display.

If you attempt to delete a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:MEASview:DElete{SUMMary|FLATness|CONStellation|EVM|MERRor|PERRor|
CRESPonse|STABle|PVTi
```

### Arguments

The following table lists the arguments.

Argument	WLAN measurement view
SUMMary	Summary measurements
FLATness	Spectral Flatness measurement
Table continued...	

Argument	WLAN measurement view
CONStellation	Constellation measurement

## Examples

`DISP:WLAN:MEAS:DEL PERR` deletes the Phase Error measurement view.

## DISPlay:WLAN:MEASview:NEW (No Query Form)

Displays a new measurement view in the WLAN display.

If you attempt to open a view that is currently displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:MEASview:NEW{SUMMary|FLATness|CONStellation|EVM|MERRor|PERRor|CRESpouse|STABle|PVTi`

## Arguments

The following table lists the arguments.

**Table 41:**

Argument	WLAN measurement view
SUMMary	Summary measurements
FLATness	Spectral Flatness measurement
CONStellation	Constellation measurement

## Examples

`DISP:WLAN:MEASVIEW:NEW CONSTELLATION` creates the Constellation measurement view.

## DISPlay:WLAN:MEASview:SElect

Selects a measurement view on the screen in the WLAN measurements. The query command returns the currently selected view.

If you attempt to select a view that is not displayed on screen, the error (-200, "Execution error; Measurement is already running") will be returned.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:MEASview:SElect {SUMMary|FLATness|CONStellation|EVM|MERRor|PERRor|CRESpouse|STABle|P`

`DISPlay:WLAN:MEASview:SElect?`

## Arguments

The following table lists the arguments.

Argument	WLAN measurement view
SUMMary	Summary measurements
FLATness	Spectral Flatness measurement
CONStellation	Constellation measurement
EVM	Error Vector Magnitude measurement
MERRor	Magnitude Error vs. Time
PERRor	Phase Error vs. Time
CRESpouse	Channel Response measurement
STABle	Symbol Table
PVTime	Power Vs. Time

## Examples

`DISP:WLAN:MEASVIEW:SEL CONSTELLATION` selects the Constellation measurement view.

`DISP:WLAN:MEASVIEW:SEL?` might return FLAT, indicating that the Spectral Flatness measurement view is the currently selected view.

## DISPlay:WLAN:MERRor:FREQuency:AUTO (No Query Form)

Automatically rescales the horizontal subcarrier (frequency) start and stop values to fit the waveform to the screen in the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:MERRor:FREQuency:AUTO`



## Arguments

None

## Examples

`DISPlay:WLAN:MERRor:FREQuency:AUTO` automatically rescales the horizontal Subcarrier (frequency) values to fit the waveform to the screen.

## DISPlay:WLAN:MERRor:FREQuency:RESet (No Query Form)

Automatically resets the horizontal subcarrier (frequency) start and stop values for the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:FREQuency:RESet
```

## Arguments

None

## Examples

`DISP:WLAN:MERR:FREQ:RES` automatically resets the horizontal subcarrier start and stop values for the WLAN Magnitude Error display.

## DISPlay:WLAN:MERRor:FREQuency:START

Specifies or queries the horizontal start value (left edge) on the WLAN Magnitude Error display. The units are either Frequency (Hz) or Subcarriers, and are set using the command [\[SENSe\]:WLAN:UNIT:FREQuency](#).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:FREQuency:START <NRf>
```

```
DISPlay:WLAN:MERRor:FREQuency:START?
```

## Arguments

Floating point number that specifies the horizontal start value (left edge) on the graph.

## Returns

The returned value indicates the Subcarrier (for non-b standards only) or frequency shown at the left edge of the display.

## Examples

`DISP:WLAN:MERR:FREQ:START 2.41e9` sets the horizontal start value (left edge) to 2.41e9 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:MERRor:FREQuency:STOP

Specifies or queries the horizontal stop value (right edge) on the WLAN Magnitude Error display. The units are either Frequency (Hz) or Subcarriers, and are set using the command `[SENSe]:WLAN:UNIT:FREQuency`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:MERRor:FREQuency:STOP <Nrf>`

`DISPlay:WLAN:MERRor:FREQuency:STOP?`

## Arguments

Floating point number that specifies the horizontal stop value (right edge) on the display.

## Examples

`DISP:WLAN:MERR:FREQ:STOP 8.1250` sets the horizontal stop value to 8.1250 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:MERRor:MARKer:SHOW:STATE

Shows or hides the marker readout on the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:MARKer:SHOW:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:MERRor:MARKer:SHOW:STATe?
```

## Arguments

ON or 1 specifies to show the marker readout.

OFF or 0 specifies to turn the marker readout off.

## Examples

```
DISP:WLAN:MERR:MARK:SHOW:STATE ON specifies to turn the marker readout on .
```

## DISPlay:WLAN:MERRor[:SCALE]:AUTO (No Query Form)

Automatically adjusts all vertical and horizontal values to provide the best display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor[:SCALE]:AUTO
```

## Arguments

None

## Examples

```
DISP:WLAN:MERR:SCALE:AUTO automatically adjusts all vertical and horizontal values to provide the best display.
```

## DISPlay:WLAN:MERRor:TIME:AUTO (No Query Form)

Sets the horizontal start and stop values to fit the waveform to the WLAN Magnitude Error display, and also enables Auto horizontal scaling so the waveform will remain properly scaled as burst lengths change.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:MERRor:TIME:AUTO

## Arguments

None

## Examples

DISP:WLAN:MERR:TIME:AUTO sets the horizontal start and stop values to fit the waveform to the display, and also enables Auto horizontal scaling.

## DISPlay:WLAN:MERRor:TIME:AUTO:STATe

When enabled, automatically sets the horizontal symbol scale value for the WLAN MERR display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:MERRor:TIME:AUTO:STATe {0|1|OFF|ON}

DISPlay:WLAN:MERRor:TIME:AUTO:STATe?

## Arguments

ON or 1 sets the horizontal symbol scale value automatically.

OFF or 0 specifies to stop setting the horizontal symbol scale value automatically.

## Examples

DISP:WLAN:MERR:TIME:AUTO:STATE ON specifies to the horizontal symbol scale value automatically.

## DISPlay:WLAN:MERRor:TIME:RESet (No Query Form)

Resets the horizontal symbol start and stop values for the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:TIME:RESet
```

## Arguments

None

## Examples

DISP:WLAN:MERR:TIME:RES resets the horizontal symbol start and stop values for the WLAN Magnitude Error display.

## DISPlay:WLAN:MERRor:TIME:START

Sets or queries the horizontal symbol start value for the WLAN Magnitude Error display. The units are either Symbols or Seconds, and are set using the command [SENSe]:WLAN:UNIT:TIME.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:TIME:START <Nrf>
```

```
DISPlay:WLAN:MERRor:TIME:START?
```

## Arguments

Floating point number.

## Examples

DISP:WLAN:MERR:TIME:START 40 sets the horizontal symbol start value of the WLAN Magnitude Error display to 40 Symbols (assuming the units have been set to Symbols).

## DISPlay:WLAN:MERRor:TIME:STOP

Sets or queries the horizontal symbol stop value for the WLAN Magnitude Error display. The units are either Symbols or Seconds, and are set using the command [SENSe]:WLAN:UNIT:TIME.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:TIME:STOP <NRf>
```

```
DISPlay:WLAN:MERRor:TIME:STOP?
```

## Arguments

Floating point number.

## Examples

DISP:WLAN:MERR:TIME:STOP 804.96e-6 sets the horizontal symbol stop value for the WLAN Magnitude Error display to 804.960  $\mu$ s (assuming the units have been set to Seconds).

## DISPlay:WLAN:MERRor:WINDow:SElect:PLOT

Specifies or queries which graphs are displayed in the WLAN Magnitude Error display. Select from Symbols, Subcarriers, or Both.

To set the Time units, use [\[SENSE\]:WLAN:UNIT:TIME](#). To set the Frequency units, use [\[SENSE\]:WLAN:UNIT:FREQuency](#). To set the Radix, use [\[SENSE\]:WLAN:RADix](#).



**Note:** Use this command only with multicarrier standards and not singlecarrier (802.11b).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:WINDow:SElect:PLOT{SYMBols|SCARier|BOTH}
```

```
DISPlay:WLAN:MERRor:WINDow:SElect:PLOT?
```

## Arguments

SYMBols displays the Symbols graph.

SCARier displays the Subcarriers graph.

BOTH displays both graphs.

## Examples

DISP:WLAN:MERR:WIND:SEL:PLOT SCAR displays the Subcarrier graph on the WLAN Magnitude Error display.

## DISPlay:WLAN:MERRor:WINDow:TRACe:GRATicule:GRID:STATe

Shows or hides the graticule grid on the display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:WINDow:TRACe:GRATicule:GRID:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:MERRor:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

ON or 1 specifies to show the graticule.

OFF or 0 specifies to turn it off.

## Examples

```
DISP:WLAN:MERR:WIND:TRAC:GRAT:GRID:STATE ON specifies to turn the graticule on.
```

## DISPlay:WLAN:MERRor:Y:AUTO (No Query Form)

Automatically sets the vertical scale and position values for the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:Y:AUTO
```

## Arguments

None

## Examples

```
DISP:WLAN:MERR:Y:AUTO automatically sets the vertical scale and position values for the WLAN Magnitude Error display.
```

## DISPlay:WLAN:MERRor:Y:OFFSet

Sets or queries the offset from the center (vertical position), in percent terms, for the WLAN Magnitude Error display. To set the vertical scale, use the command `DISPlay:WLAN:EVM:Y[:SCALE]`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:Y:OFFSet <NRf>
```

```
DISPlay:WLAN:MERRor:Y:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the Magnitude Error plot at least partially visible.

on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

## Returns

The offset from the center of the Y axis for the Magnitude Error plot as a floating point number.

## Examples

```
DISP:WLAN:MERR:Y:OFFS 150 sets the offset from the center of the vertical axis of the Magnitude Error plot to 150 %.
```

## DISPlay:WLAN:MERRor:Y:RESet (No Query Form)

Resets the vertical scale and position (offset) for the WLAN Magnitude Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:MERRor:Y:RESet
```

## Arguments

None

## Examples

```
DISP:WLAN:MERR:Y:RES specifies to reset the vertical scale and position (offset) values for the WLAN Magnitude Error display.
```



## DISPlay:WLAN:MERRor:Y[:SCALE]

Sets or queries the vertical scale value for the WLAN Magnitude Error display. To set the vertical position (offset) value, use the command DISPlay:WLAN:EVM:Y:OFFSet.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:MERRor:Y[:SCALE] <NRf>
```

```
DISPlay:WLAN:MERRor:Y[:SCALE]?
```

### Arguments

Floating point number that specifies the vertical scale value for the WLAN Magnitude Error display.

### Examples

```
DISP:WLAN:MERR:Y:SCALE 10 sets the vertical scale for the WLAN Magnitude Error display to 10.
```

## DISPlay:WLAN:PERRor:FREQuency:AUTO (No Query Form)

Automatically rescales the horizontal subcarrier (frequency) start and stop values to fit the waveform to the screen in the WLAN Phase Error display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PERRor:FREQuency:AUTO
```

### Arguments

None

### Examples

```
DISP:WLAN:PERR:FREQ:AUTO automatically rescales the horizontal Subcarrier (frequency) start and stop values to fit the waveform to the screen.
```

## **DISPlay:WLAN:PERRor:FREQuency:RESet (No Query Form)**

Automatically resets the horizontal subcarrier (frequency) start and stop values for the WLAN Phase Error display.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PERRor:FREQuency:RESet
```

### **Arguments**

None

### **Examples**

`DISP:WLAN:PERR:FREQ:RES` automatically resets the horizontal Subcarrier start and stop values for the WLAN Phase Error display.

## **DISPlay:WLAN:PERRor:FREQuency:START**

Specifies or queries the horizontal start value (left edge) on the WLAN Phase Error display. The units are either Frequency (Hz) or Subcarriers, and are set using the command `[SENSe]:WLAN:UNIT:FREQuency`.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PERRor:FREQuency:START <NRf>
```

```
DISPlay:WLAN:PERRor:FREQuency:START?
```

### **Arguments**

Floating point number that specifies the horizontal start value (left edge) on the display.

### **Returns**

The readout indicates the Subcarrier (for non-b standards only) or frequency shown at the left edge of the display.

## Examples

`DISP:WLAN:PERR:FREQ:START -8.1250` sets the horizontal start value to -8.1250 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:PERRor:FREQuency:STOP

Specifies or queries the horizontal stop value (right edge) on the WLAN Phase Error display. The units are either Frequency (Hz) or Subcarriers, and are set using the command `[SENSe]:WLAN:UNIT:FREQuency`.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:PERRor:FREQuency:STOP <NRf>`

`DISPlay:WLAN:PERRor:FREQuency:STOP?`

### Arguments

Floating point number that specifies the horizontal stop value (right edge) on the display.

## Examples

`DISP:WLAN:PERR:FREQ:STOP 8.1250` sets the horizontal stop value to 8.1250 MHz (assuming the units have been set to Frequency).

## DISPlay:WLAN:PERRor:MARKer:SHOW:STATe

Show or hides the Marker readout on the WLAN Phase Error display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:PERRor:MARKer:SHOW:STATe {0|1|OFF|ON}`

`DISPlay:WLAN:PERRor:MARKer:SHOW:STATe?`

## Arguments

ON or 1 specifies to show the Marker readout.

OFF or 0 specifies to turn the Marker readout off.

## Examples

DISP:WLAN:PERR:MARK:SHOW:STATE ON specifies to turn the Marker readout on.

## DISPlay:WLAN:PERRor[:SCALE]:AUTO (No Query Form)

Automatically adjusts all vertical and horizontal settings to provide the best display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:PERRor[:SCALE]:AUTO

## Arguments

None

## Examples

DISP:WLAN:PERRor:SCALE:AUTO automatically adjusts all vertical and horizontal settings to provide the best display.

## DISPlay:WLAN:PERRor:TIME:AUTO (No Query Form)

Automatically sets the horizontal symbol start and stop values to fit the waveform to the screen for the WLAN Phase Error vs. Symbol plot. In addition, sets the state for this feature to ON.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:PERRor:TIME:AUTO

## Arguments

None

## Examples

`DISP:WLAN:PERR:TIME:AUTO` sets the instrument to automatically select the horizontal symbol start and stop values to fit the waveform to the screen, and turns the AUTO feature on.

## DISPlay:WLAN:PERRor:TIME:AUTO:STATE

When enabled, automatically sets the horizontal symbol scale value for the WLAN Phase Error display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:TIME:AUTO:STATE {0|1|OFF|ON}
```

```
DISPlay:WLAN:PERRor:TIME:AUTO:STATE?
```

## Arguments

`ON` or `1` specifies to set the Analysis Time value automatically.

`OFF` or `0` specifies to turn stop setting the Analysis Time value automatically.

## Examples

`DISP:WLAN:PERR:TIME:AUTO:STATE ON` specifies to set the Analysis Time value automatically, based on the requirements of the selected display

## DISPlay:WLAN:PERRor:TIME:RESet (No Query Form)

Resets the horizontal symbol start and stop values for the WLAN Phase Error Average vs. Symbol display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:TIME:RESet
```

## Arguments

None

## Examples

`DISP:WLAN:PERR:TIME:RES` resets the horizontal symbol start and stop values for the WLAN Phase Error Average vs. Symbol display.

## DISPlay:WLAN:PERRor:TIME:START

Sets or queries the horizontal symbol start value for the WLAN Phase Error Average vs. Symbol display. The units are either Symbols or Seconds, and are set using the command `[SENSe]:WLAN:UNIT:TIME`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

`DISPlay:WLAN:PERRor:TIME:START <NRf>`

`DISPlay:WLAN:PERRor:TIME:START?`

## Arguments

Floating point number.

## Examples

`DISP:WLAN:PERR:TIME:START 40` sets the horizontal symbol start value of the Average vs. Symbol display to 40 Symbols (assuming the units have been set to Symbols).

## DISPlay:WLAN:PERRor:TIME:STOP

Sets or queries the horizontal symbol stop value for the WLAN Phase Error Average vs. Symbol display. The units are either Symbols or Seconds, and are set using the command `[SENSe]:WLAN:UNIT:TIME`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:TIME:STOP <NRf>
```

```
DISPlay:WLAN:PERRor:TIME:STOP?
```

## Arguments

Floating point number.

## Examples

`DISP:WLAN:PERR:TIME:STOP 804.960` sets the horizontal symbol stop value for the Average vs. Symbol display to 804.960  $\mu$ s (assuming the units have been set to Seconds).

## DISPlay:WLAN:PERRor:WINDow:SElect:PLOT

Specifies or queries which displays are displayed in the WLAN Phase Error analysis display (Symbols, Subcarriers, or Both).

To set the Time units, use [\[SENSe\]:WLAN:UNIT:TIME](#). To set the Frequency units, use [\[SENSe\]:WLAN:UNIT:FREQuency](#). To set the Radix, use [\[SENSe\]:WLAN:RADix](#).



**Note:** Use this command only with multicarrier standards and not singlecarrier (802.11b).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:WINDow:SElect:PLOT {SYMBols|SCARier|BOTH}
```

```
DISPlay:WLAN:PERRor:WINDow:SElect:PLOT?
```

## Arguments

`SYMBols` displays the Symbols graph.

`SCARier` displays the Subcarriers graph.

`BOTH` displays both graphs.

## Examples

`DISP:WLAN:PERR:WIND:SEL:PLOT SCAR` displays the Subcarrier graph on the WLAN Phase Error analysis display.

## DISPlay:WLAN:PERRor:WINDow:TRACe:GRATicule:GRID:STATe

Shows or hides the graticule grid on the screen.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:WINDow:TRACe:GRATicule:GRID:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:PERRor:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

ON or 1 specifies to show the graticule.

OFF or 0 specifies to turn it off.

## Examples

DISP:WLAN:PERR:WIND:TRAC:GRAT:GRID:STATE ON specifies to turn the graticule on.

## DISPlay:WLAN:PERRor:Y:AUTO (No Query Form)

Automatically selects the vertical scale and position values for the WLAN Phase Error Symbol and Subcarrier plots.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:Y:AUTO
```

## Arguments

None

## Examples

DISP:WLAN:PERR:Y:AUTO automatically selects the vertical scale and position values for the WLAN Phase Error Symbol and Subcarrier plots.

## DISPlay:WLAN:PERRor:Y:OFFSet

Sets or queries the value of the offset from the center (vertical position), in percent terms, for the WLAN Phase Error Symbol and Subcarrier plots. To set the Vertical Scale, use the command DISPlay:WLAN:PERRor:Y[:SCALE].



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:Y:OFFSet <NRf>
```

```
DISPlay:WLAN:PERRor:Y:OFFSet?
```

## Arguments

Floating point number between limits which are set as a function of the current scale value guaranteed to keep the plot at least partially visible on the display. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up. 0 re-centers the plot.

## Returns

The value of the offset from the center of the Y axis for the Phase Error plot as a floating point number.

## Examples

```
DISP:WLAN:PERR:Y:OFFS 150 sets the offset from the center (vertical position) of the Phase Error plot to 150 %.
```

## DISPlay:WLAN:PERRor:Y:RESet (No Query Form)

Resets the vertical scale and position (offset) values for the WLAN Phase Error Symbol and Subcarrier plots.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PERRor:Y:RESet
```

## Arguments

None

## Examples

```
DISP:WLAN:PERR:Y:RES specifies to reset the vertical scale and position (offset) values for the WLAN Phase Error plot.
```

## **DISPlay:WLAN:PERRor:Y[:SCALE]**

Sets or queries the vertical scale value for the WLAN Phase Error plots. To set the vertical position (offset) value, use the command `DISPlay:WLAN:PERRor:Y:OFFSet`.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PERRor:Y[:SCALE] <Nrf>
```

```
DISPlay:WLAN:PERRor:Y[:SCALE]?
```

### **Arguments**

Floating point number that specifies the vertical scale value.

### **Examples**

```
DISP:WLAN:PERR:Y:SCALE 10 sets the vertical scale value to 10.
```

## **DISPlay:WLAN:PVTime:BURSt:X[:SCALE]**

Sets or queries the value of the scale (width), in seconds, for the WLAN Power vs. Time display when using horizontal Full Burst view.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PVTime:BURSt:X[:SCALE] <Nrf>
```

```
DISPlay:WLAN:PVTime:BURSt:X[:SCALE]?
```

### **Arguments**

Floating point number that represents the value of the scale (width), in seconds, when using horizontal Full Burst view.

### **Examples**

```
DISP:WLAN:PVT:BURS:X:SCALE 5.0E-6 sets the width of the display to 5.000  $\mu$ S for the horizontal Full Burst view.
```

## DISPlay:WLAN:PVTTime:BURSt:X[:SCALe]:AUTO (No Query Form)

Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Full Burst view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTTime:BURSt:X[:SCALe]:AUTO
```

### Arguments

None

### Examples

DISP:WLAN:PVT:BURSt:X:SCALE:AUTO automatically sets the starting time (position) and scale (width) values for the best display.

## DISPlay:WLAN:PVTTime:BURSt:X[:SCALe]:OFFSet

Sets or queries the starting time (position) value, in seconds, for the WLAN Power vs. Time display when using the horizontal Full Burst view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTTime:BURSt:X[:SCALe]:OFFSet <NRf>
```

```
DISPlay:WLAN:PVTTime:BURSt:X[:SCALe]:OFFSet?
```

### Arguments

Floating point number that represents the starting time (position) value, in seconds, when using horizontal Full Burst view.

### Examples

DISP:WLAN:PVT:BURSt:X:OFFSet 5.0E-6 sets the starting time (position) of the graph to 5.000  $\mu$ S for horizontal Full Burst view.

## **DISPlay:WLAN:PVTime:BURSt:X[:SCALe]:RESet (No Query Form)**

Resets the starting time (position) and scale (width) values for the horizontal Full Burst view on the WLAN Power vs. Time display.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PVTime:BURSt:X[:SCALe]:RESet
```

### **Arguments**

None

### **Examples**

`DISP:WLAN:PVT:BURST:X:SCALE:RESET` resets the starting time (position) and scale (width) values for the horizontal Full Burst view.

## **DISPlay:WLAN:PVTime:FALL:X[:SCALe]**

Sets or queries the scale (width) value, in seconds, for the WLAN Power vs. Time display when using the horizontal Falling Edge view.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Display commands

### **Syntax**

```
DISPlay:WLAN:PVTime:FALL:X[:SCALe] <NRf>
```

```
DISPlay:WLAN:PVTime:FALL:X[:SCALe]?
```

### **Arguments**

Floating point number that represents the scale (width), in seconds, when using the horizontal Falling Edge view.

### **Examples**

`DISP:WLAN:PVT:FALL:X:SCALE 5.0E-6` sets the width of the graph to 5.000  $\mu$ S when using the horizontal Falling Edge view.

## DISPlay:WLAN:PVTime:FALL:X[:SCALE]:AUTO (No Query Form)

Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Falling Edge view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTime:FALL:X[:SCALE]:AUTO
```

### Arguments

None

### Examples

DISP:WLAN:PVT:FALL:X:SCALE:AUTO automatically sets the starting time (position) and scale (width) values for the best display.

## DISPlay:WLAN:PVTime:FALL:X[:SCALE]:OFFSet

Sets or queries the value for the starting time (offset) of the graph, in seconds, for the WLAN Power vs. Time display when using the horizontal Falling Edge view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTime:FALL:X[:SCALE]:OFFSet <NRf>
```

```
DISPlay:WLAN:PVTime:FALL:X[:SCALE]:OFFSet?
```

### Arguments

Floating point number that represents the value of the starting time (offset) of the graph, in seconds, when using the horizontal Falling Edge view.

## Examples

`DISP:WLAN:PVT:FALL:X:SCALE 5.0` sets the starting time (position) of the graph to 5.000  $\mu$ S when using the horizontal Falling Edge view.

## **DISPlay:WLAN:PVTime:FALL:X[:SCALE]:RESet (No Query Form)**

Resets the starting time (position) and scale (width) values for the horizontal Falling Edge view on the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:FALL:X[:SCALE]:RESet
```

## Arguments

None

## Examples

`DISP:WLAN:PVT:FALL:X:SCALE:RESET` resets the starting time (position) and scale (width) values for the horizontal Falling Edge view.

## **DISPlay:WLAN:PVTime:MARKer:SHOW:STATe**

Shows or hides the marker readout on the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:MARKer:SHOW:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:PVTime:MARKer:SHOW:STATe?
```

## Arguments

`ON` or `1` specifies to show the marker readout.

`OFF` or `0` specifies to turn the marker readout off.

## Examples

`DISP:WLAN:PVT:MARK:SHOW:STATE ON` specifies to turn the marker readout on.

## DISPlay:WLAN:PVTime:RISE:X[:SCALE]

Sets or queries the scale (width) value, in seconds, for the WLAN Power vs. Time display when using the horizontal Rising Edge view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:PVTime:RISE:X[:SCALE] <NRf>`

`DISPlay:WLAN:PVTime:RISE:X[:SCALE]?`

### Arguments

Floating point number that represents the scale (width) value, in seconds, for the horizontal Rising Edge view.

## Examples

`DISP:WLAN:PVT:RISE:X:SCALE 5.0E-6` sets the width of the graph to 5.000µS for the horizontal Rising Edge view.

## DISPlay:WLAN:PVTime:RISE:X[:SCALE]:AUTO (No Query Form)

Automatically sets the starting time (position) and scale (width) values for the best display in the horizontal Rising Edge view of the WLAN Power vs. Time display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:PVTime:RISE:X[:SCALE]:AUTO`

### Arguments

None

## Examples

`DISP:WLAN:PVT:RISE:X:SCALE:AUTO` automatically sets the starting time (position) and scale (width) values for the best display.

## **DISPlay:WLAN:PVTime:RISE:X[:SCALE]:OFFSet**

Sets or queries the starting time value (offset), in seconds, for the WLAN Power vs. Time display when using the horizontal Rising Edge view.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTime:RISE:X[:SCALE]:OFFSet <NRf>
```

```
DISPlay:WLAN:PVTime:RISE:X[:SCALE]:OFFSet?
```

### Arguments

Floating point number that represents the value of the starting time (offset) of the graph, in seconds, for the horizontal Rising Edge view.

## Examples

`DISP:WLAN:PVT:RISE:X:OFFSET 5.0e-6` sets the starting time (position) of the graph to 5.000  $\mu$ S for the horizontal Rising Edge view.

## **DISPlay:WLAN:PVTime:RISE:X[:SCALE]:RESet (No Query Form)**

Resets the starting time (position) and scale (width) values for the horizontal Rising Edge view on the WLAN Power vs. Time display.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

```
DISPlay:WLAN:PVTime:RISE:X[:SCALE]:RESet
```

### Arguments

None



## Examples

`DISP:WLAN:PVT:RISE:X:SCALE:RESET` resets the starting time (position) and scale (width) values for the horizontal Rising Edge view.

## DISPlay:WLAN:PVTime:WINDow:SElect:PLOT

Specifies or queries which view to use for the WLAN Power vs. Time display. Select from Full Burst, Rising Edge or Falling Edge.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

### Syntax

`DISPlay:WLAN:PVTime:WINDow:SElect:PLOT {BURSt|RISE|FALL}`

`DISPlay:WLAN:PVTime:WINDow:SElect:PLOT?`

### Arguments

`BURSt` sets the view to Full Burst, which displays the entire packet, with vertical lines indicating the length of the burst.

`RISE` sets the view to Rising Edge, which zooms the display into the interval around the burst rising edge, with vertical lines indicating the 10% to 90% Power-On Ramp time.

`FALL` sets the view to Falling Edge, which zooms the display into the interval around the burst Falling Edge, with vertical lines indicating the 90% to 10% Power-Down Ramp time.

### Returns

`DISP:WLAN:PVT:WIND:SEL:PLOT BURSt` sets the view to Full Burst.

`DISP:WLAN:PVT:WIND:SEL:PLOT?` might return `RISE`, indicating that the view has been set to Rising Edge.

## DISPlay:WLAN:PVTime:WINDow:TRACe:GRATicule:GRID:STATe

Shows or hides the graticule grid on the screen.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:WINDow:TRACe:GRATicule:GRID:STATe {0|1|OFF|ON}
```

```
DISPlay:WLAN:PVTime:WINDow:TRACe:GRATicule:GRID:STATe?
```

## Arguments

ON or 1 specifies to show the graticule.

OFF or 0 specifies to turn it off.

## Examples

```
DISP:WLAN:PVT:WIND:TRAC:GRAT:GRID:STATE ON specifies to turn the graticule on.
```

## DISPlay:WLAN:PVTime:Y[:SCALE]

Sets or queries the vertical scale value for the WLAN Power vs. Time display.

To set the vertical position (offset) value, use the command [DISPlay:WLAN:PVTime:Y\[:SCALE\]:OFFSet](#).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:Y[:SCALE] <NRf>
```

```
DISPlay:WLAN:PVTime:Y[:SCALE]?
```

## Arguments

Floating point number that specifies the vertical scale value for the WLAN Power vs. Time display.

## Examples

```
DISP:WLAN:PVT:Y:SCALE 10 sets the vertical scale value to 10 dB.
```

## DISPlay:WLAN:PVTime:Y[:SCALE]:AUTO (No Query Form)

Automatically selects the vertical scale and position values for the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

`DISP:WLAN:PVT:Y:AUTO` automatically selects the vertical scale and position values for the display.

## DISPlay:WLAN:PVTime:Y[:SCALE]:OFFSet

Sets or queries the value of the vertical offset (top edge of the vertical axis) for the WLAN Power vs. Time display. To set the vertical scale, use the command `DISPlay:WLAN:PVTime:Y[:SCALE]`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:Y[:SCALE]:OFFSet <NRf>
```

```
DISPlay:WLAN:PVTime:Y[:SCALE]:OFFSet?
```

## Arguments

Floating point number between limits which specifies the value of the vertical offset. Positive inputs push the plot down (as viewed by the user) and negative inputs push the plot up.

## Returns

The value of the vertical offset for the WLAN Power vs. Time display as a floating point number.

## Examples

`DISP:WLAN:PVT:Y:OFFS -10` sets the vertical offset to -10 dBm.

## DISPlay:WLAN:PVTime:Y[:SCALE]:PDIVision

Sets or queries the value of the vertical scale for the WLAN Power vs. Time display, in dB/division. This is only a visual control for panning the graph.

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:Y[:SCALE]:PDIVision <NRf>
```

```
DISPlay:WLAN:PVTime:Y[:SCALE]:PDIVision?
```

## Arguments

Floating point value that represents the value of the vertical scale, in dB/division, for the WLAN Power vs. Time display.

## Examples

```
DISP:WLAN:PVT:Y:SCALE:PDIV 20 sets the vertical scale to 20.0 dB/division.
```

## DISPlay:WLAN:PVTime:Y[:SCALE]:RESet (No Query Form)

Resets the vertical scale and position (offset) values for the WLAN Power vs. Time display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

```
DISPlay:WLAN:PVTime:Y[:SCALE]:RESet
```

## Arguments

None

## Examples

```
DISP:WLAN:PVT:Y:RES specifies to reset the vertical scale and position (offset) values for the WLAN Power vs. Time display.
```

## DISPlay:WLAN:SUMMARY:EVMUnits

Sets or queries the EVM Units setting in the WLAN Summary display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Display commands

## Syntax

DISPlay:WLAN:SUMMary:EVMUnits {PERCent | DB} DISPlay:WLAN:SUMMary:EVMUnits?

## Arguments

PERCent. Units in the WLAN Summary display are shown as percentages.

DB. Units in the WLAN Summary display are shown in decibels (dB).

## \*ESE

Sets or queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). Refer to Section 3, *Status and Events*, for the register information.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

\*ESE <value>

\*ESE?

## Related Commands

\*CLS, \*ESR?, \*SRE, \*STB?

## Arguments

<value>: : <NR1> is a value in the range from 0 through 255.

The binary bits of the ESER are set according to this value.

## Examples

\*ESE 145 sets the ESER to binary 10010001, which enables the PON, EXE, and OPC bits.

\*ESE? might return the string \*ESE 184, showing that the ESER contains the binary value 10111000.

## \*ESR? (Query Only)

Returns the contents of the Standard Event Status Register (SESR). \*ESR? also clears the SESR (since reading the SESR clears it). Refer to Section 3, *Status and Events*, for the register information.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

\*ESR?

## Related Commands

\*CLS, \*ESE, \*SRE, \*STB?

## Arguments

None

## Returns

<NR1> representing the contents of the SESR by a 0 to 255 decimal number.

## Examples

\*ESR? might return the value 213, showing that the SESR contains binary 11010101.

# Fetch commands

## FETCh:ACPower? (Query Only)

Returns the Channel power and ACPR measurement results for all available channels.

## Conditions

Measurement views: Channel power and ACPR

## Group

Fetch commands

## Syntax

FETCh:ACPower?

## Arguments

None

## Returns

<chan\_power>,<acpr\_lower (1)>,<acpr\_upper (1)>,  
<acpr\_lower (2)>,<acpr\_upper (2)>,...  
<acpr\_lower (n)>,<acpr\_upper (n)>

Where

`<chan_power>` is the average power of the main channel as the power reference in dBm. The unit can be changed by the `[SENSe]:POWer:UNITs` command.

`<acpr_lower (n)>` is the ACPR for the lower channel #n in dB.

`<acpr_upper (n)>` is the ACPR for the upper channel #n in dB.

The number of n depends on the setting of the `[SENSe]:ACPower:CHANnel:PAIRs` command.

## Examples

`FETCH:ACPOWER?` might return 4.227,-28.420,-23.847,-22.316,-29.225, indicating (average power of the main channel) = 4.227 dBm, (ACPR for the lower channel 1) = -28.420 dB, (ACPR for the upper channel 1) = -23.847 dB, (ACPR for the lower channel 2) = -22.316 dB, and (ACPR for the upper channel 2) = -29.225 dB.

## FETCh:ACPower:CHANnel:POWer? (Query Only)

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Fetch commands

### Syntax

`FETCh:ACPower:CHANnel:POWer?`

### Arguments

None

### Returns

`<chan_power>::<NRf>` is the average power of the main channel in dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCH:ACPOWER:CHANNEL:POWER?` might return 4.227, indicating that the average power of the main channel is 4.227 dBm.

## FETCh:ACPower:SPECTrum? (Query Only)

Returns spectrum trace data of the Channel power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Fetch commands

## Syntax

FETCh:ACPower:SPECTrum?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCH:ACPOWER:SPECTRUM? might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

## FETCh:{AM|FM|PM}? (Query Only)

Returns the trace data in the AM/FM/PM measurement.

## Conditions

Measurement views: AM, FM, and PM

## Group

Fetch commands

## Syntax

FETCh:{AM|FM|PM}?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the modulation factor in percent (AM), frequency deviation in Hz (FM), or phase deviation in degrees (PM) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.



## Examples

`FETCH:AM?` might return `#3156xxxx...` (156-byte data) for the AM measurement trace.

## FETCh:AM:AMINdex? (Query Only)

Returns the modulation depth which is the (positive peak modulation factor - negative peak modulation factor)/2 returned as a percentage (%).

### Conditions

Measurement views: AM

### Group

Fetch commands

### Syntax

`FETCh:AM:AMINdex?`

### Arguments

None

### Returns

`<value>::<NRf>` the modulation index.

## Examples

`FETCH:AM:AMINDEX?` might return `77.1854035556E-3`, indicating the modulation index is 0.0772% or 77.2 m%.

## FETCh:AM:AMNegative? (Query Only)

Returns the negative peak modulation factor (–AM) in the AM measurement.

### Conditions

Measurement views: AM

### Group

Fetch commands

### Syntax

`FETCh:AM:AMNegative?`

### Arguments

None

## Returns

<-AM> : : <NRf> is the negative peak modulation factor in percent (%).

## Examples

FETCH:AM:AMNEGATIVE? might return -23.4, indicating the negative peak modulation factor is -23.4%.

## FETCh:AM:AMPositive? (Query Only)

Returns the positive peak modulation factor (+AM) in the AM measurement.

## Conditions

Meas

urement views: AM

## Group

Fetch commands

## Syntax

FETCh:AM:AMPositive?

## Arguments

None

## Returns

<+AM> : : <NRf> is the positive peak modulation factor in percent (%).

## Examples

FETCH:AM:AMPOSITIVE? might return 43.8, indicating the positive peak modulation factor is 43.8%.

## FETCh:AM:RESult? (Query Only)

Returns the AM measurement results.

## Conditions

Measurement views: AM

## Group

Fetch commands

## Syntax

FETCh:AM:RESult?

## Arguments

None

## Returns

<+AM>, <-AM>, <Total AM>

Where

<+AM> :: <NRf> is the positive peak modulation factor in percent (%).

<-AM> :: <NRf> is the negative peak modulation factor in percent (%).

<Total AM> :: <NRf> is the (peak to peak modulation factor)/2 in percent (%).

## Examples

FETCH:AM:RESULT? might return 62.63,—50.89,56.76.

## FETCh:AUDio:FERRor? (Query Only)

Returns the carrier frequency error in the audio measurement. This command is only available for FM and PM measurements.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

FETCh:AUDio:FERRor?

## Arguments

None

## Returns

<value> ::= <NRf> the carrier frequency error.

## Examples

FETCH:AUDIO:FERROR? might return 419.9529809622 indicating that the audio frequency error is approximately 420 Hz.

## FETCh:AUDio:FREQuency? (Query Only)

Returns the audio frequency.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:FREQuency?`

## Arguments

None

## Returns

`<value>` ::= `<NRf>` the audio frequency.

## Examples

`FETCH:AUDIO:FREQUENCY?` might return `5.8239462705E+3` indicating that the audio frequency is 5.824 kHz.

## FETCh:AUDio:HARMonic:COUNT? (Query Only)

Returns the number of harmonics in the audio spectrum measurement.

The number of harmonics appear in the results table in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Fetch commands

## Syntax

`FETCh:AUDio:HARMonic:COUNT?`

## Related Commands

`FETCh:AUDio:NHARmonic:COUNT?`

## Arguments

None

## Returns

`<count>` ::= `<NR1>` the number of harmonics in the measurement view.

## Examples

`FETCH:AUDIO:HARMONIC:COUNT?` might return 3 indicating that there are three harmonics in the measurement.

## FETCh:AUDio:HARMonic<x>:AMPLitude? (Query Only)

Returns the amplitude of the specified harmonic in the audio spectrum measurement.

The value <x> represents the specific harmonic, from 1 to the value specified by the CALCulate:AUDio:HARMonic:HNUMBER command.

### Conditions

Measurement views: Audio Spectrum

### Group

Fetch commands

### Syntax

```
FETCh:AUDio:HARMonic<x>:AMPLitude?
```

### Related Commands

```
CALCulate:AUDio:HARMonic:HNUMBER
```

### Arguments

None

### Returns

<value> ::= <NRf> the amplitude of the specified harmonic in dBc.

### Examples

FETCh:AUDio:HARMonic3:AMPLitude? might return -2.861 indicating that amplitude of the third harmonic is -2.86 dBc.

## FETCh:AUDio:HARMonic<x>:FREQuency? (Query Only)

Returns the frequency of the specified harmonic in the audio spectrum measurement.

The value <x> represents the specific harmonic, from 1 to the value specified by the CALCulate:AUDio:HARMonic:HNUMBER command.

### Conditions

Measurement views: Audio Spectrum

### Group

Fetch commands

### Syntax

```
FETCh:AUDio:HARMonic<x>:FREQuency?
```

### Related Commands

```
CALCulate:AUDio:HARMonic:HNUMBER
```

## Arguments

None

## Returns

<value> ::= <NRF> the frequency of the specified harmonic.

## Examples

`FETCh:AUDio:HARMonic3:FREQuency?` might return `17.4718007813E+3` indicating that the frequency of the third harmonic is 17.472 kHz.

## FETCh:AUDio:HNOise? (Query Only)

Returns the difference between the current RMS modulation value and the reference value.

This information is only available when the Hum & Noise is measurement is enabled; see the `[SENSe]:AUDio:HNOise:ENABle` command. The reference value is stored after capturing the reference by pressing the Capture Reference button or by enabling the `[SENSe]:AUDio:HNOise:REFerence` command.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:HNOise?`

## Related Commands

`[SENSe]:AUDio:HNOise:ENABle` `[SENSe]:AUDio:HNOise:REFerence`

## Arguments

None

## Returns

<value> ::= <NRF> specifies the difference between the Hum & Noise reference and the actual signal in dB.

## Examples

`FETCh:AUDio:HNOISE?` might return `0.00` dB indicating that there is no difference between the Hum & Noise reference and the actual signal.

## FETCh:AUDio:HNREFerence? (Query Only)

Returns the Hum and Noise RMS modulation reference value.

The value is stored after capturing the reference by pressing the Capture Reference button or by enabling the `[SENSE]:AUDIO:HNOise:REference` command. This information is only available when the Hum & Noise measurement is enabled; see the `[SENSE]:AUDIO:HNOise:ENABLE` command.

For AM signal types, the reference represents modulation depth. For FM signal types, the reference represents frequency deviation. For PM signal types the reference represents phase. For Direct signal types the reference represents power.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:HNREference?
```

## Related Commands

```
[SENSe]:AUDio:HNOise:ENABLE [SENSe]:AUDio:HNOise:REference
```

## Arguments

None

## Returns

<value> ::= <NRf> the Hum & Noise reference frequency.

## Examples

`FETCH:AUDIO:HNREFERENCE?` might return `20.581121E+3` indicating that the Hum & Noise reference frequency is 20.58 kHz.

## FETCh:AUDio:HPTPeak? (Query Only)

Returns the half peak-to-peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:HPTPeak?
```

## Arguments

None

## Returns

<value> ::= <NRF> specifies the half peak modulation excursion.

## Examples

`FETCH:AUDIO:HPTPEAK?` might return `125.28231E+3` indicating that the half peak modulation excursion is 125.28 kHz.

## FETCh:AUDio:MODDist? (Query Only)

Returns the modulation distortion for the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:MODDist?`

## Arguments

None

## Returns

<value> ::= <NRF> specifies the modulation distortion.

## Examples

`FETCH:AUDIO:MODDIST?` might return `98.6282113` indicating that the modulation distortion is 98.628%.

## FETCh:AUDio:NHARmonic:COUNT? (Query Only)

Returns the number of non-harmonics in the audio spectrum measurement.

The number of non-harmonics appear in the results table in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Fetch commands



## Syntax

```
FETCh:AUDio:NHARmonic:COUNT?
```

## Related Commands

```
FETCh:AUDio:HARMonic:COUNT?
```

## Arguments

None

## Returns

<count> ::= <NR1> the number of non-harmonics in the measurement view.

## Examples

FETCH:AUDIO:NHARMONIC:COUNT? might return 7 indicating that there are seven harmonics in the measurement.

## FETCh:AUDio:NHARmonic<x>:AMPLitude? (Query Only)

Returns the amplitude of the specified non-harmonic in the audio spectrum measurement.

The value <x> represents the specific non-harmonic, from 1 to the value specified by the CALCulate:AUDio:HARMonic:NHNumber command.

## Conditions

Measurement views: Audio Spectrum

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:NHARmonic<x>:AMPLitude?
```

## Related Commands

```
CALCulate:AUDio:HARMonic:NHNumber
```

## Arguments

None

## Returns

<value> ::= <NRf> the amplitude of the specified non-harmonic in dBc.

## Examples

FETCh:AUDio:NHARMonic3:AMPLitude? might return -2.861 indicating that amplitude of the third non-harmonic is -2.86 dBc.

## **FETCh:AUDio:NHARmonic<x>:FREQuency? (Query Only)**

Returns the frequency of the specified non-harmonic in the audio spectrum measurement.

The value <x> represents the specific non-harmonic, from 1 to the value specified by the `CALCulate:AUDio:HARMonic:NHNumber` command.

### **Conditions**

Measurement views: Audio Spectrum

### **Group**

Fetch commands

### **Syntax**

```
FETCh:AUDio:NHARmonic<x>:FREQuency?
```

### **Related Commands**

```
CALCulate:AUDio:HARMonic:NHNumber
```

### **Arguments**

None

### **Returns**

<value> ::= <NRF> the frequency of the specified non-harmonic.

### **Examples**

`FETCh:AUDio:NHARmonic3:FREQuency?` might return `17.4718007813E+3` indicating that the frequency of the third non-harmonic is 17.472 kHz.

## **FETCh:AUDio:PNEGative? (Query Only)**

Returns the minus peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

### **Conditions**

Measurement views: Audio Summary

### **Group**

Fetch commands

### **Syntax**

```
FETCh:AUDio:PNEGative?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the minus peak modulation excursion.

## Examples

`FETCH:AUDIO:PNEGATIVE?` might return `-196.04321E+3` indicating that the minus peak modulation excursion is `-196.04` kHz.

## FETCh:AUDio:POWer? (Query Only)

Returns the carrier power or signal power depending the signal type.

If the signal type is Direct, the returned value is the average power of the input signal. If the signal type is AM, FM, or PM, the returned value is the carrier power; the average power of the carrier signal with the modulation removed.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:POWer?`

## Arguments

None

## Returns

<value> ::= <NRf> specifies the carrier or signal power in dBm.

## Examples

`FETCH:AUDIO:POWER?` might return `-22.231123` indicating that the carrier power is `-22.23` dBm.

## FETCh:AUDio:PPOSitive? (Query Only)

Returns the positive peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is “% Modulation Depth.” For FM signal types, the modulation excursion is “Frequency Deviation.” For PM signal types, the modulation excursion is “Phase Deviation.” For Direct signal types, there is no modulation excursion; it is actually “signal excursion.”

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:PPOSitive?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the positive peak modulation excursion.

## Examples

`FETCh:AUDio:PPOSITIVE?` might return `215.04321E+3` indicating that the positive peak modulation excursion is `-215.04` kHz.

## FETCh:AUDio:RMS? (Query Only)

Returns the RMS modulation excursion for the audio measurement (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:RMS?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the RMS modulation excursion.

## Examples

`FETCh:AUDio:RMS?` might return `20.575039E+3` indicating that RMS modulation excursion is `20.575` kHz.

## FETCh:AUDio:SINad? (Query Only)

Returns the signal-to-noise and distortion for the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:SINad?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the RMS modulation excursion.

## Examples

FETCh:AUDIO:SINAD? might return 176.229024E-3 indicating that the signal-to-noise and distortion for the audio measurement is 0.18 dB.

## FETCh:AUDio:SNOise? (Query Only)

Returns the signal level to noise level with the harmonic distortion and non-harmonic distortion components removed.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:SNOise?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the signal level to noise level.

## Examples

FETCh:AUDIO:SNOISE? might return -12.8156364 indicating the signal level to noise level is -12.82 dB.

## FETCh:AUDio:SPECTrum:TRACe<x>? (Query Only)

Returns the audio spectrum trace data for the audio measurement.

The <x> represents trace 1 for this measurement. No other traces are supported.

## Conditions

Measurement views: Audio Spectrum

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:SPECTrum:TRACe<x>?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><date(1)><data(2)> . . . <data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating point format specified in IEEE.488.2.

## Examples

FETCh:AUDio:SPECTrum:TRACe1? might return #516004xxxx ... (16004 byte data) for the audio measurement.

## FETCh:AUDio:THDist? (Query Only)

Returns the percentage of the total harmonic distortion in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

```
FETCh:AUDio:THDist?
```

## Related Commands

```
FETCh:AUDio:THDist:DB?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the percentage of the total harmonic distortion.

## Examples

`FETCH:AUDIO:THDIST?` might return 53.332921 indicating the percentage of total harmonic distortion is 53.333%.

## FETCh:AUDio:THDist:DB? (Query Only)

Returns the total harmonic distortion in dB in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:THDist:DB?`

## Related Commands

`FETCh:AUDio:THDist?`

## Arguments

None

## Returns

<value> ::= <NRf> specifies the total harmonic distortion in dB.

## Examples

`FETCH:AUDIO:THDIST:DB?` might return -5.46009 indicating that the total harmonic distortion is -5.46 dB.

## FETCh:AUDio:TNHDist? (Query Only)

Returns the percentage of the total non-harmonic distortion in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:TNHDist?`

## Related Commands

`FETCh:AUDio:TNHDist:DB?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the percentage of the total non-harmonic distortion.

## Examples

`FETCh:AUDIO:TNHDIST?` might return 297.332921 indicating the percentage of total non-harmonic distortion is 297.333%.

## FETCh:AUDio:TNHDist:DB? (Query Only)

Returns the total non-harmonic distortion in dB in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Fetch commands

## Syntax

`FETCh:AUDio:TNHDist:DB?`

## Related Commands

`FETCh:AUDio:TNHDist?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the total non-harmonic distortion in dB.

## Examples

`FETCh:AUDIO:TNHDIST:DB?` might return 9.46009 indicating that the total harmonic distortion is 9.46 dB.

## FETCh:AVTime[:MEASview<y>]:AVERage? (Query Only)

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.



## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:AVERage?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<avg>: : <NRf> is the RMS amplitude in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:AVTIME:MEASVIEW1:AVERAGE? might return -2.53, indicating the RMS amplitude is -2.53 dBm for Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:{FIRSt|SECond|THIRd|FOURth}? (Query Only)

Returns the trace data in the Amplitude versus Time measurement.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCH:AVTIME:MEASVIEW1:FIRST? might return #3156xxx... (156-byte data) for Trace 1 for the Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SELEct command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:MAXimum?
```

## Related Commands

```
FETCh:AVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max>::<NRf> is the maximum Amplitude in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCH:AVTIME:MEASVIEW1:MAXIMUM? might return -2.84, indicating the maximum amplitude is -2.84 dBm for the Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SELEct command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:MAXLocation?
```

## Related Commands

```
FETCh:AVTime[:MEASview<y>]:MAXimum?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max\_time>::<Nrf> is the time at the maximum in seconds.

## Examples

FETCH:AVTIME:MEASVIEW1:MAXLOCATION:FIRST? might return 25.03E-9, indicating the amplitude is maximum at 25.03 ns in Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:MINimum? (Query Only)

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:MINimum?
```

## Related Commands

```
FETCh:AVTime[:MEASview<y>]:MINLocation?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

`<min>::<NRf>` is the minimum amplitude in dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCh:AVTIME:MEASVIEW1:MINIMUM?` might return `-57.64`, indicating the minimum amplitude is `-57.64` dBm for Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:MINLocation? (Query Only)

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
FETCh:AVTime[:MEASview<y>]:MINimum?
```

## Arguments

The parameter `<y>=1` to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Returns

`<min_time>::<NRf>` is the time at the minimum in seconds.

## Examples

`FETCh:AVTIME:MEASVIEW1:MINLOCATION?` might return `450.7E-9`, indicating the amplitude is minimum at `450.7` ns for Amplitude versus Time Measurement View 1.

## FETCh:AVTime[:MEASview<y>]:RESult? (Query Only)

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
FETCh:AVTime[:MEASview<y>]:RESult?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
<max>,<max_time>,<min>,<min_time>,<rms>
```

Where

<max>::<NRf> is the maximum amplitude in dBm.

<max\_time>::<NRf> is the time at the maximum in seconds.

<min>::<NRf> is the minimum amplitude in dBm.

<min\_time>::<NRf> is the time at the minimum in seconds.

<rms>::<NRf> is the RMS amplitude in dBm.

The unit of amplitude can be changed by the [SENSE]:POWER:UNITs command.

## Examples

FETCH:AVTIME:MEASVIEW1:RESULT? might return -2.68,48.62E-6,-82.47,22.11E-6,-8.24, indicating that the maximum amplitude is -2.68 dBm at 48.62  $\mu$ s, the minimum amplitude is -82.47 dBm at 22.11  $\mu$ s, and the RMS amplitude is -8.24 dBm on Amplitude versus Time Measurement View 1.

## :FETCh:AVTime[:MEASview<y>]:AVERage:{FIRSt|SECond|THIRd|FOURth}

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the :TRACe<x>:AVTime[:MEASview<y>]:SElect command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
:FETCh:AVTime[:MEASview<y>]:AVERage:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max> : : <NRf> is the maximum Amplitude in dBm.

The unit can be changed by the [SENSe] : POWer : UNITS command

## Examples

: FETCh : AVTIME : MEASVIEW1 : AVERAGE : FIRST? might return -2.53, indicating the RMS amplitude is -2.53 dBm for Trace1 in Amplitude versus Time Measurement View 1.

## **:FETCh:AVTime[:MEASview<y>]:MAXimum:{FIRSt|SECond|THIRd|FOURth}**

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

: FETCh : AVTIME [ : MEASview<y> ] : MAXimum : { FIRSt | SECond | THIRd | FOURth } ?

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max> : : <NRf> is the maximum Amplitude in dBm.

The unit can be changed by the [SENSe] : POWer : UNITS command.

## Examples

: FETCh : AVTIME : MEASVIEW1 : MAXIMUM : FIRST? might return -2.84, indicating the maximum amplitude is -2.84 dBm for the Trace1 in Amplitude versus Time Measurement View 1.

## **:FETCh:AVTime[:MEASview<y>]:MAXLocation:{FIRSt|SECond|THIRd|FOURth}**

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the :TRACe<x>:AVTime[:MEASview<y>]:SElect command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
:FETCh:AVTime[:MEASview<y>]:MAXimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max>: : <NRf> is the maximum Amplitude in dBm.

## Examples

:FETCh:AVTIME:MEASVIEW1:MAXLOCATION? might return 25.03E-9, indicating the amplitude is maximum at 25.03 ns in Trace1 for Amplitude versus Time Measurement View 1.

## :FETCh:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the :TRACe<x>:AVTime[:MEASview<y>]:SElect command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

```
:FETCh:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max>: : <NRf> is the maximum Amplitude in dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`:FETCH:AVTIME:MEASVIEW1:MINIMUM:FIRST?` might return `-57.64`, indicating the minimum amplitude is `-57.64` dBm for Trace1 in Amplitude versus Time Measurement View 1.

## **:FETCh:AVTime[:MEASview<y>]:MINLocation:{FIRSt|SECond|THIRd|FOURth}**

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the `:TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

The mnemonics `FIRSt`, `SECond`, `THIRd`, and `FOURth` represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time

## Group

Fetch commands

## Syntax

`:FETCh:AVTime[:MEASview<y>]:MINLocation:{FIRSt|SECond|THIRd|FOURth}?`

## Related Commands

`:FETCh:AVTime[:MEASview<y>]:MINimum?`

## Arguments

The parameter `<y>=1` to `8` represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is `1`

## Returns

`<max>::<NRf>` is the maximum Amplitude in dBm.

## Examples

`:FETCH:AVTIME:MEASVIEW1:MINLOCATION:FIRST?` Might return `450.7E-9`, indicating the amplitude is minimum at `450.7` ns for Trace1 in Amplitude versus Time Measurement View 1.

## **:FETCh:AVTime[:MEASview<y>]:RESult:{FIRSt|SECond|THIRd|FOURth}**

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the `:TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

The mnemonics `FIRSt`, `SECond`, `THIRd`, and `FOURth` represent Trace 1, Trace 2, Trace 3, and Math trace, respectively.

## Conditions

Measurement views: Amplitude versus Time



## Group

Fetch commands

## Syntax

```
:FETCh:AVTime[:MEASview<y>]:RESult:{FIRSt|SECOnd|THIRd|FOURth}??
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
<max>,<max_time>,<min>,<min_time>,<rms>
```

Where

<max>::<NRf> is the maximum amplitude in dBm.

<max\_time>::<NRf> is the time at the maximum in seconds.

<min>::<NRf> is the minimum amplitude in dBm.

<min\_time>::<NRf> is the time at the minimum in seconds.

<rms>::<NRf> is the RMS amplitude in dBm.

The unit of amplitude can be changed by the [:SENSE]:POWER:UNITs command.

## Examples

:FETCh:AVTIME:MEASVIEW1:RESult:FIRSt? might return -2.68,48.62E-6,-82.47,22.11E-6,-8.24, indicating that the maximum amplitude is -2.68 dBm at 48.62  $\mu$ s, the minimum amplitude is -82.47 dBm at 22.11  $\mu$ s, and the RMS amplitude is -8.24 dBm for Trace1 on Amplitude versus Time Measurement View 1.

## FETCh:BIBEmissions:FTX? (Query Only)

Returns the band number corresponding to the transmitted FTX detected in the Bluetooth InBand Emission display.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Fetch commands

## Syntax

```
FETCh:BIBEmissions:FTX?
```

## Arguments

None

## Returns

<value> ::= <NR1> the band number.

## Examples

FETCH:BIBEMISSIONS:FTX? might return 39 indicating the band number corresponding to transmitted FTX.

## FETCh:BIBEmIssions:POWer? (Query Only)

Returns all channel power results from the Bluetooth InBand Emission display.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Fetch commands

## Syntax

FETCh:BIBEmIssions:POWer?

## Arguments

None

## Returns

<Band0\_power>,<Band1\_power>,<Band2\_power>, . . . <Band78\_power>

Where:

Bandx\_power is the integrated power in that band. All power values are in dBm

## Examples

FETCH:BIBEMISSIONS:POWER? might return 4.227,-28.420,-23.847, . . .,-29.225, indicating:

(integrated power for Band 0) = 4.227 dBm,

(Integrated power for Band 1) = -28.420 dBm,

(Integrated power for Band 2) = -23.847 dBm, and

(Integrated power for Band 78) = -29.225 dBm.

## FETCh:BIBEmIssions:RESUlts:STATUs? (Query Only)

Returns the Pass/Fail result status of the Bluetooth InBand Emission display.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Fetch commands

## Syntax

```
FETCh:BIbEmissions:RESUlts:STATUs?
```

## Arguments

None

## Examples

`FETCh:BIbEMISSIONS:RESULTS:STATUS?` returns the Pass/Fail status result on the display.

## FETCh:BLUEtooth:CONSte:FERRor? (Query Only)

Returns the frequency error in Hz in the Bluetooth Constellation. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:CONSte:FERRor?
```

## Arguments

None

## Returns

`<freq_error> ::= <NRf>` the frequency error.

## Examples

`FETCh:BLUEtooth:CONSte:FERRor?` Might return `-10.7E+3`, which is a frequency error of `-10.7 kHz`.

## FETCh:BLUEtooth:CONSte:TRACe? (Query Only)

Queries the Bluetooth Constellation trace data.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:CONSTE:TRACe?

## Arguments

None

## Returns

#<num\_digit><num\_byte><l(1)><Q1><l(2)><Q2> . . . <l(n)><Qn>

Where:

<num\_digit> is the number of bytes in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<l(n)> and <Qn> are the normalized I- and Q-coordinate values at the nth data point. 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:BLUETOOTH:CONSTE:TRACe? might return #43848xxxx . . . (3848-byte data) for the trace.

## FETCH:BLUETOOTH:EDIagram:FDEVIation? (Query Only)

Returns the frequency deviation vs. time trace data with the X values in the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth eye diagram.

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:EDIagram:FDEVIation?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<Y(1)> and <X(1)> are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

`FETCh:BLUEtooth:EDIagram:FDEVIiation?` might return `#3160xxx...` (160-byte data) for the frequency deviation versus time trace.

## FETCh:BLUEtooth:EDIagram:FERRor? (Query Only)

Returns the frequency error in the Bluetooth eye diagram measurement.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:EDIagram:FERRor?`

### Arguments

None

### Returns

`<freq_error> ::= <NRf>` the frequency error in Hz.

## Examples

`FETCh:BLUEtooth:EDIagram:FERRor?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

## FETCh:BLUEtooth:EDIagram:HORiz:SCALE? (Query Only)

Returns the value of the horizontal scale in the Bluetooth eye diagram.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:EDIagram:HORiz:SCALE?`

### Arguments

None

### Returns

`<value> ::= <NR3>` the horizontal scale value.

## Examples

`FETCH:BLUETOOTH:EDIAGRAM:HORIZ:SCALE?` might return 2 indicating the horizontal scale value is 2 symbols.

## FETCh:BLUEtooth:EDIagram:I? (Query Only)

Returns the I vs. time trace in the Bluetooth eye diagram.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:EDIagram:I?`

### Arguments

None

### Returns

`#<num_digits><num_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>`

Where

`<num_digits>` is the number of bytes in `<num_bytes>`.

`<num_bytes>` is the number of bytes of data that follow.

`<Y(1)>` and `<X(1)>` are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

`FETCh:BLUEtooth:EDIagram:I?` might return `#3160xxxx...` (160-byte data) for the I vs. time trace.

## FETCh:BLUEtooth:EDIagram:Q? (Query Only)

Returns the Q vs. time trace in the Bluetooth eye diagram.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:EDIagram:Q?`

## Arguments

None

## Returns

#<num\_digits><num\_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<Y(1)> and <X(1)> are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

FETCh:BLUEtooth:EDIagram:Q? might return #3160xxxx... (160-byte data) for the Q vs. time trace.

## FETCh:BLUEtooth:FDVTime:TRACe? (Query Only)

Returns the Bluetooth Frequency Deviation vs. Time trace data.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Fetch commands

## Syntax

FETCh:BLUEtooth:FDVTime:TRACe?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><data(1)><data(2)> . . . <data(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<data(n)> is the frequency deviation in Hz at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

FETCh:BLUEtooth:FDVTime:TRACe? might return #3160xxxx... (160-byte data) for the trace.

## **FETCh:BLUEtooth:FREQuency:ERROr? (Query Only)**

Returns the frequency error from the Bluetooth Constellation diagram.

### **Conditions**

Measurement views: Bluetooth Constellation

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:FREQuency:ERROr?

### **Arguments**

None

### **Returns**

<NR3>, the frequency error value in Hz.

### **Examples**

FETCh:BLUETOOTH:FREQUENCY:ERROR? might return 0.0000 indicating the frequency error is 0.0000 Hz.

## **FETCh:BLUEtooth:FREQuency:ERROr:TYPE? (Query Only)**

Returns the frequency error type in the Bluetooth Constellation display.

### **Conditions**

Measurement views: Bluetooth Constellation

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:FREQuency:ERROr:TYPE?

### **Arguments**

None

### **Returns**

1 indicates the error is automatically detected.

0 indicates the error is not automatically detected.



## Examples

`FETCH:BLUETOOTH:FREQUENCY:ERROR:TYPE?` might return 1 indicating the frequency error is automatically detected.

## FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:F1FZero? (Query Only)

Returns the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble) in the Bluetooth Frequency Deviation vs. Time display.

### Conditions

Measurement views: Frequency Deviation vs. Time

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:F1FZero?`

### Arguments

None

### Returns

<NR3> the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble)

## Examples

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:F1FZero?` might return -207.6465301514 indicating the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble)

## FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFN5? (Query Only)

Returns the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away in the Bluetooth Frequency Deviation vs. Time display.

### Conditions

Measurement views: Frequency Deviation vs. Time

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFN5?`

## Arguments

None

## Returns

<NR3> the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away.

## Examples

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNF5?` might return `-207.6465301514` indicating the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away.

## **FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNF5:INTERval? (Query Only)**

Returns the interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNF5:INTERval?`

## Arguments

None

## Returns

<NR1> interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred.

## Examples

`FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNF5:INTERval?` might return `16`, indicating the interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred. Here `16` refers to the 16th interval and that the drift of frequency offset calculated in the 16th interval from that calculated in the 11th interval is the maximum among those calculated 50  $\mu$ s away.

## **FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO? (Query Only)**

Returns the maximum drift of the frequency offset of the intervals in the payload from the preamble packet in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO?
```

## Arguments

None

## Returns

<NR3> the maximum drift of frequency offset calculated in payload intervals from the offset calculated in the preamble.

## Examples

FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZero? might return -207.6465301514, indicating that the maximum drift of frequency offset calculated in payload intervals from the offset calculated in the preamble is -207.6465301515 Hz.

## FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO:INTERval? (Query Only)

Returns the interval (n) at which the maximum drift  $f_n - f_0$  occurred in the Bluetooth Frequency Deviation vs. Time display. (The  $f_n - f_0$  indicates the drift of the frequency offset calculated in each payload interval from the offset calculated in the preamble.)

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZERO:INTERval?
```

## Arguments

None

## Returns

<NR1> the interval at which the maximum drift ( $f_n - f_0$ ) occurred.

## Examples

FETCh:BLUEtooth:FREQuency:OFFSet:DRIFt:FNFZero:INTERval? might return 16, indicating that at the 16th payload interval, the maximum drift occurred.  $f(16) - f(0)$  is the maximum drift among all  $f(n) - f(0)$ .

## FETCh:BLUEtooth:FREQuency:OFFSet:MAX? (Query Only)

Returns the maximum frequency offset in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time, CF Offset

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:FREQuency:OFFSet:MAX?`

## Arguments

None

## Returns

<NR3> the maximum frequency offset value.

## Examples

`FETCh:BLUEtooth:FREQuency:OFFSet:MAX?` might return -189.4632263184, the frequency offset value.

## FETCh:BLUEtooth:FREQuency:OFFSet:MAX:INTERval? (Query Only)

Returns the maximum frequency offset interval in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time, CF Offset

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:FREQuency:OFFSet:MAX:INTERval?`

## Arguments

None

## Returns

<NR1> the maximum frequency offset interval.

## Examples

`FETCh:BLUEtooth:FREQuency:OFFSet:MAX:INTERval?` might return 271 indicating that the maximum drift  $f(n) - f(0)$  occurred at the 271st payload interval.

## FETCh:BLUEtooth:FREQuency:OFFSet:PREAmble? (Query Only)

Returns the offset calculated in the preamble region in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time, CF Offset

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:FREQuency:OFFSet:PREAmble?
```

## Arguments

None

## Returns

<NR3> the offset calculated in the preamble region.

## Examples

FETCh:BLUEtooth:FREQuency:OFFSet:PREAmble? might return 106.4204711914, indicating the offset calculated in the preamble region.

## FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F? (Query Only)

Returns the average frequency value when the selected test pattern is Other in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F?
```

## Arguments

None

## Returns

<NRf> the average frequency.

## Examples

FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F? might return 140.8309531250E+3, the average frequency value.

## **FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F2F1ratio? (Query Only)**

Returns the average ratio value for the high and low frequency deviation in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage: F2F1ratio?
```

### **Arguments**

None

### **Returns**

<NRf> the average ratio value.

### **Examples**

FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:F2F1ratio? might return 24.37E-3, the average ratio value.

## **FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:FONE? (Query Only)**

Returns the average frequency deviation for the low deviation pattern in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:FONE?
```

### **Arguments**

None

### **Examples**

FETCh:BLUEtooth:MODUlation:CHARacteristics:AVErage:FONE? might return 139.8309531250E+3, the average frequency deviation for the low deviation pattern.

---

## **FETCh:BLUEtooth:MODUlation:CHARacteristics:AVERage:FTWO? (Query Only)**

Returns the average frequency deviation for the high deviation pattern in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:MODUlation:CHARacteristics:AVERage:FTWO?

### **Arguments**

None

### **Examples**

FETCh:BLUEtooth:MODUlation:CHARacteristics:AVERage:FTWO? might return 139.8309531250E+3, the average frequency deviation for the high deviation pattern.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:COUNT? (Query Only)**

Returns the number of packets analyzed to obtain average drift results from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:COUNT?

### **Arguments**

None

### **Returns**

<NR1> the number of packets analyzed.

### **Examples**

FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:COUNT? might return 8, the number of packets analyzed.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero? (Query Only)**

Returns the drift f1–f0 from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero?

### **Arguments**

None

### **Returns**

<NRf> the drift f1–f0 value.

### **Examples**

FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero? might return 2.846E+3 indicating that the drift f1–f0 value is 2.846 kHz.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero:STATUs? (Query Only)**

Returns the Pass/Fail drift f1–f0 result status in the packet from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero: STATUs?

### **Arguments**

None

### **Returns**

<string> the Pass/Fail result status.



## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:F1FZERO:STATUS?` might return PASS, FAIL, or NA indicating whether the drift f1–f0 is within the specified limit.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFN5? (Query Only)**

Returns the maximum drift of the fn–fn-5 value from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFN5?`

### Arguments

None

### Returns

<NRf> the fn–fn-5 value.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5?` might return 2.846E+3 indicating that the maximum drift fn–fn-5 value in the preamble region is -2.846 kHz.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFN5:STATUs? (Query Only)**

Returns the Pass/Fail status of the maximum fn–fn-5 drift in the packet from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFN5:STATUs?`

### Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum drift  $f_n-f_{n-5}$  is within the specified limit.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFZero? (Query Only)**

Returns the maximum drift  $f_n-f_0$  from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFZero?`

## Arguments

None

## Returns

<NRf> the  $f_n-f_{n0}$  value.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZERO?` might return 4.846E+3 indicating that the drift  $f_n-f_0$  value in the preamble region is 4.846 kHz.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFZero:STATUs? (Query Only)**

Returns the Pass/Fail result status of the  $f_n-f_0$  maximum drift in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:FNFZero: STATUs?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZERO:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum drift  $f_n - f_0$  is within the specified limit.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:CODIng:SCHEme (Query Only)**

Returns the offset drift coding scheme value (LE Coded) in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded: CODIng:SCHEme?`

## Arguments

None

## Returns

<NR1> 2 or 8

Where 2 is data at 500 Kb/s and 8 is 125 Kb/s

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:CODING:SCHEME` might return "2", indicating the offset drift coding scheme value is 2.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F0FTThree (Query Only)**

Returns the Drift value  $f_0 - f_3$  for LE Coded in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE?`

## Arguments

None

## Returns

<NRf> Maximum Frequency value in kHz

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE` might return 207.2 kHz, indicating the f0-f3 drift value is 207.2 kHz .

## **FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS (Query Only)**

Returns the Drift value Status of f0-f3 for LE Coded in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS?`

## Arguments

None

## Returns

<String> PASS ,FAIL or NA

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS` might return Pass, indicating the test result Status of f0-f3 was Pass.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour (Query Only)**

Returns the Drift value of f1-f4 for LE Coded in Bluetooth Summary display.

### **Conditions**

Measurement view: Bluetooth Summary display

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded: F1FFour?
```

### **Arguments**

None

### **Returns**

<NRf> Maximum Frequency value in kHz

### **Examples**

FETCh:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F1FFOUR might return -10.22 kHz, indicating the f1-f4 drift value is -10.22 kHz.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour:STATUs (Query Only)**

Returns the Drift value Status of f1-f4 for LE Coded in Bluetooth Summary display

### **Conditions**

Measurement view: Bluetooth Summary display

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded: F1FFour:STATUs?
```

### **Arguments**

None

### **Returns**

<String> PASS ,FAIL or NA

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F1FFOUR:STATUS` might return Pass, indicating the test result Status of f1-f4 was Pass.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:LECoded:FNFN3 (Query Only)**

Returns the Drift value of fn-f(n-3) for LE Coded in Bluetooth Summary display

### Conditions

Measurement view: Bluetooth Summary display

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:LECoded: FNFN3?`

### Arguments

None

### Returns

<NRf> Maximum Frequency value in kHz

### Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3` might return -10.42 kHz, indicating the fn-f(n-3) drift value is -10.42 kHz.

## **FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:LECoded:FNFN3:STATUs (Query Only)**

Returns the Drift value Status of fn-f(n-3) for LE Coded in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:FREQuency:OFFSet:DRIFt:LECoded: FNFN3:STATUs?`

## Arguments

None

## Returns

<String> PASS ,FAIL or NA

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3:STATUS` might return Pass, indicating the test result Status of fn-f(n-3) was Pass.

## FETCh:BLUEtooth:SUMMArY:FREQUency:OFFSet:MAX? (Query Only)

Returns the maximum frequency offset present in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:FREQUency:OFFSet:MAX?`

## Arguments

None

## Returns

<NRf> the maximum frequency offset value in the packet.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX?` might return 7.446E+3 indicating that the maximum frequency offset value in the packet is 7.446 kHz.

## FETCh:BLUEtooth:SUMMArY:FREQUency:OFFSet:MAX:STATus? (Query Only)

Returns the Pass/Fail result status of the frequency offset maximum in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:MAX:STATus?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum frequency offset in the packet is within the specified limit.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:PREAmble? (Query Only)**

Returns the frequency offset value in the preamble region from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:PREAmble?`

## Arguments

None

## Returns

<NRf> the frequency offset value in the preamble region.

## Examples

`FETCH:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAMBLE?` might return 3.546E+3 indicating that the frequency offset value in the preamble region is 3.546 kHz.

## **FETCh:BLUEtooth:SUMMary:FREQuency:OFFSet:PREAmble:STATus? (Query Only)**

Returns the Pass/Fail result status of the frequency offset in the preamble region.



## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:FREQUency:OFFSet:PREAmble:STATus?
```

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

FETCh:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble:STATUS? might return PASS, FAIL, or NA indicating whether the frequency offset in the preamble region is within the specified limit.

## FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:F:MAX:PERCentage:COUNT? (Query Only)

Returns the number of packets analyzed out of 10 packets for  $\Delta F$  average from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVErage:DELTA:F:MAX:PERCentage:COUNT?
```

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

FETCh:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:F:MAX:PERCENTAGE:COUNT? might return 3 indicating that three packets were analyzed.

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:DELTA:FONE:MAX:PERCentage:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for the  $\Delta F1$ avg from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:DELTA:FONE:MAX:PERCentage:COUNT?
```

### **Arguments**

None

### **Returns**

<NR1> the number of packets analyzed.

### **Examples**

```
FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FONE:MAX:PERCENTAGE:COUNT? might return 6 indicating that six packets were analyzed.
```

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:DELTA:FTWO:MAX:PERCentage:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for the  $\Delta F2$ Max% from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:DELTA:FTWO:MAX:PERCentage:COUNT?
```

### **Arguments**

None

## Returns

<NR1> the number of packets analyzed.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:COUNT?` might return 6 indicating that six packets were analyzed.

## **FETCh:BLUEtooth:SUMMArY:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentage? (Query Only)**

Returns the percentage of the  $\Delta F2Max$  values that are above the specified limit.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentage?`

## Arguments

None

## Returns

<NRf> the number of packets analyzed.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE?` might return 99.2 indicating that 99.2% of the deltaF2 maximum values are above the specified limit.

## **FETCh:BLUEtooth:SUMMArY:MODUlation:CHARacteristics:AVErage:DELTA:FTWO:MAX:PERCentage:STATUs? (Query Only)**

Returns the Pass/Fail result status of the  $\Delta F2Max\%$  field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:DELTA:FTWO:MAX:PERCentage:STATUS?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:STATUS?` might return PASS, FAIL, or NA indicating the Pass/Fail status of the  $\Delta F_{2Max}$  field in the Bluetooth summary.

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F? (Query Only)**

Returns the  $\Delta F_{avg}$  value from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage: F?`

## Arguments

None

## Returns

<NRf> the delta F value.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F?` might return 2.8723E+3 indicating the delta F average is 2.8723 kHz.

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F2F1ratio? (Query Only)**

Returns the ratio of the  $\Delta F_{2avg}$  to  $\Delta F_{1avg}$  from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage: F2F1ratio?
```

## Arguments

None

## Returns

<NRf> the value of the delta F1 to delta F2 ratio.

## Examples

FETCh:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1RATIO? might return 225.12E+3 indicating the average ratio is 22.5.12 kHz.

## FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F2F1ratio:STATus? (Query Only)

Returns the Pass/Fail results status of the  $\Delta F2_{avg}/\Delta F1_{avg}$  field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage: F2F1ratio:STATus?
```

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

FETCh:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1RATIO:STATUS? might return PASS, FAIL, or NA indicating the Pass/Fail result status of the  $\Delta F2_{avg}/\Delta F1_{avg}$  field.

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:FONE? (Query Only)**

Returns the  $\Delta F_{1avg}$  value from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage: FONE?
```

### **Arguments**

None

### **Returns**

<NRf> the delta F1 value.

### **Examples**

FETCh:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FONE? might return 155.3672E+3 indicating the delta F1 average is 155.3672 kHz.

## **FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:FONE:STATu s? (Query Only)**

Returns the Pass/Fail results status of the  $\Delta F_{1avg}$  from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Fetch commands

### **Syntax**

```
FETCh:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage: FONE:STATUS?
```

### **Arguments**

None

### **Returns**

<string> the Pass/Fail results status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FONE:STATUS?` might return PASS, FAIL, or NA indicating whether the delta F1 average is within the limit.

## **FETCh:BLUEtooth:SUMMArY:MODUlation:CHARacteristics:AVErage:FTW0? (Query Only)**

Returns the  $\Delta F_{2avg}$  value from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Fetch commands

### Syntax

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE: FTW0?`

### Arguments

None

### Returns

<NRf> the delta F2 value.

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FTW0?` might return 225.12E+3 indicating that the delta F2 value is 225.12 kHz.

## **FETCh:BLUEtooth:SUMMArY:MODUlation:CHARacteristics:INDex (Query Only)**

Returns the Modulation index value in Bluetooth Summary display (for LE 1M, LE 2M, LE Coded).

### Conditions

Measurement view: Bluetooth Summary display

### Group

Fetch commands

### Syntax

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:INDEX?`

### Arguments

None

## Returns

<String>

## Examples

`FETCH:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:INDEX` might return "Stable", indicating the status of modulation index is "Stable".

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:AVErAge? (Query Only)**

Returns the average output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:AVErAge?`

## Arguments

None

## Returns

<NRf> the average output power value in dBm.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:AVERAGE?` might return -29.1 indicating that the average output power is -29.1 dBm.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:AVErAge:STATUs? (Query Only)**

Returns the Pass/Fail results status of the average output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:AVErAge:STATUs?`



## Arguments

None

## Returns

<string> the Pass/Fail results status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:AVERAGE:STATUS?` might return PASS, FAIL, or NA indicating whether the average output power is within the specified limit.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for the calculation of the output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:COUNT?`

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:COUNT?` might return 6 indicating that six packets were analyzed.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for calculation of the output power for EDR from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:COUNT?`

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:COUNT?` might return 6 indicating that six packets were analyzed.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:DPSK? (Query Only)**

Returns the power in the DPSK portion of the EDR burst from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:DPSK?`

## Arguments

None

## Returns

<NRf> the power in the DPSK portion of the EDR burst.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:DPSK?` might return -29.2 indicating that the average output power is -29.2 dBm.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:GPSK? (Query Only)**

Returns the power in the GFSK portion of the EDR burst from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWER:EDR:GPSK?`

## Arguments

None

## Returns

<NRf> the power in the GFSK portion of the EDR burst.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:GPSK?` might return -29.1 indicating that the average output power is -29.1 dBm.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:PEAK? (Query Only)**

Returns the peak output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:PEAK?`

## Arguments

None

## Returns

<NRf> the peak output power value in dBm.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK?` might return -29.3 indicating that the peak output power is -29.3 dBm.

## **FETCh:BLUEtooth:SUMMArY:OUTPut:POWEr:PEAK:STATUs? (Query Only)**

Returns the Pass/Fail results status of the peak output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK:STATus?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`FETCH:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK:STATUS?` might return PASS, FAIL, or NA indicating whether the peak output power is within the specified limit.

## **FETCh:BLUEtooth:SUMMary:PACKet:BDR:CRC? (Query Only)**

Returns the Basic Rate value of the CRC field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:PACKet:BDR:CRC?`

## Arguments

None

## Returns

<string> the value of the CRC field.

## Examples

`FETCh:BLUEtooth:SUMMary:PACKet:Basic Rate:CRC?` might return "0x9DB0" indicating the value of the CRC field in hexadecimal.

## **FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:ARQN? (Query Only)**

Returns the value of ARQN field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

---

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:ARQN?
```

## Arguments

None

## Returns

<NR2> the value of the ARQN field.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:Basic Rate:HEADer:ARQN? might return 0.000000 indicating the value of the flow field.

## FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:FLOW? (Query Only)

Returns the value of Flow field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:FLOW?
```

## Arguments

None

## Returns

<NR2> the flow value.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:FLOW? might return 1.000000 indicating the value of the flow field.

## FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:HEC? (Query Only)

Returns the value of HEC field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:HEC?

## Arguments

None

## Returns

<string> the value of the HEC field.

## Examples

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:HEC? might return "00110010" indicating the value of the HEC field in binary.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:LT:ADDRESS? (Query Only)

Returns the LT address from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:LT:ADDRESS?

## Arguments

None

## Returns

<string> the LT address.

## Examples

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:LT:ADDR? might return "101" indicating the LT address in binary.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:HEADER:SEQN? (Query Only)

Returns the value of SEQN field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

---

## Group

Fetch commands

## Syntax

FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:SEQN?

## Arguments

None

## Returns

<NR2> the value of the SEQN field.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:SEQN? might return 1.000000 indicating the value of the flow field.

## FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:TYPE? (Query Only)

Returns the value of Type field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:TYPE?

## Arguments

None

## Returns

<string> the value of the type field.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:BDR:HEADer:TYPE? might return "00001111" indicating the type field in binary.

## FETCh:BLUEtooth:SUMMary:PACKet:BDR:PAYLoad:LENgth? (Query Only)

Returns the Basic Rate payload length from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:PAYLOAD:LENGTH?

## Arguments

None

## Returns

<string> the value of the payload length.

## Examples

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:PAYLOAD:LENGTH? might return "0101010011" indicating the payload length value in binary.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:PREAmble? (Query Only)

Returns the preamble of the Basic Rate packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:PREAmble?

## Arguments

None

## Returns

<string> the preamble of the packet.

## Examples

FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:PREAmble? might return "0101" indicating the preamble is 0101 in binary.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BDR:SYNC:WORD? (Query Only)

Returns the sync word from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary



## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BDR:SYNC:WORD?
```

## Arguments

None

## Returns

<string> the sync word.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:BDR:SYNC:WORD? might return "Ox4F36F2CEE85390CB" indicating the sync word in hexadecimal format.

## FETCh:BLUEtooth:SUMMary:PACKet:BDR:TYPE? (Query Only)

Returns the packet type of the Basic Rate signal from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BDR:TYPE?
```

## Arguments

None

## Returns

<string> the packet type.

## Examples

FETCh:BLUEtooth:SUMMary:PACKet:BDR:TYPE? might return "DH5" indicating that the DH5 packet is being analyzed.

## FETCh:BLUEtooth:SUMMary:PACKet:BLE:ACCEss:ADDRess? (Query Only)

Returns the 32-bit Access Address from the Bluetooth Summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:ACCESS:ADDRESS?`

## Arguments

None

## Returns

<string> the 32-bit Access Address.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:ACCESS:ADDRESS?` might return "0x71764129" indicating the Access Address in hexadecimal.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:CRC? (Query Only)

Returns the value of the CRC field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:CRC?`

## Arguments

None

## Returns

<string> the value of the CRC field.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:CRC?` might return "0xAAAAAA" indicating value CRC field in hexadecimal.

## FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:CHANNEL:SELECT (Query Only)

Returns the channel selected under decoded packet information section in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:CHANnel: SElect?
```

## Arguments

None

## Returns

<String> 0 or 1 or -1 or “-”

0 and 1 represents valid

-1 represents invalid

“-” represents NIL

## Examples

FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:CHANNEL:SELECT might return “1”, indicating the selected channel is valid.

## FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:LENGth? (Query Only)

Returns the value of the length field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

```
FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:LENGth?
```

## Arguments

None

## Returns

<string> the value of the length field.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LENGTH?` might return "100101" indicating value length field in binary.

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:LLID (Query Only)**

Returns the LLID value (2 bits) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:LLID?`

### Arguments

None

### Returns

<String> Either 00,01,10,11

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LLID` might return 01, indicating the value of LLID is "01".

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:MD (Query Only)**

Returns the MD value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Fetch commands

### Syntax

`FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:MD?`

### Arguments

None

## Returns

<Boolean> 0, 1

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:MD` might return 1, indicating the value of MD is "1".

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:NESN (Query Only)**

Returns the NESN value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:NESN?`

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:NESN` might return 1, indicating the value of NESN is "1".

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:RX:ADDResS? (Query Only)**

Returns the Rx address from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:RX:ADDResS?`

## Arguments

None

## Returns

<string> the Rx address.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:RX:ADDRESS?` might return 0.00000000 indicating Rx Address.

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:SN (Query Only)**

Returns the SN value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:SN?`

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:SN` might return 1, indicating the value of SN is "1".

## **FETCh:BLUEtooth:SUMMArY:PACKet:BLE:PDU:HEADer:TX:ADDResS? (Query Only)**

Returns the Tx address from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

---

## Syntax

`FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:TX:ADDRess?`

## Arguments

None

## Returns

<string> the Tx address.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TX:ADDRESS?` might return 0.00000000 indicating Tx Address.

## **FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:TYPE? (Query Only)**

Returns the PDU type from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:PACKet:BLE:PDU:HEADer:TYPE?`

## Arguments

None

## Returns

<string> the PDU type.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TYPE?` might return "0010" indicating the PDU type in binary.

## **FETCh:BLUEtooth:SUMMary:PACKet:BLE:PREAble? (Query Only)**

Returns the preamble of the BLE packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:PACKet:BLE:PREAmble?`

## Arguments

None

## Returns

<string> the preamble of the packet.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:PREAMBLE?` might return "01010101" indicating the preamble is 01010101 in binary.

## FETCh:BLUEtooth:SUMMary:PACKet:BLE:TYPE? (Query Only)

Returns the packet type from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Fetch commands

## Syntax

`FETCh:BLUEtooth:SUMMary:PACKet:BLE:TYPE?`

## Arguments

None

## Returns

<string> the BLE packet type.

## Examples

`FETCH:BLUETOOTH:SUMMARY:PACKET:BLE:TYPE?` might return "BLE\_TEST" indicating the BLE test signal is being analyzed.

## FETCh:CCDF? (Query Only)

Returns the CCDF measurement results.

## Conditions

Measurement views: CCDF



## Group

Fetch commands

## Syntax

`FETCh:CCDF?`

## Related Commands

## Arguments

None

## Returns

`<avg_ampl>`, `<avg_ccdf>`, `<crest_factor>`, `<ampl_10>`, `<ampl_1>`,  
`<ampl_p1>`, `<ampl_p01>`, `<ampl_p001>`, `<ampl_p0001>`

Where

`<avg_ampl>` is the average amplitude in dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

`<avg_ccdf>` is the average CCDF in percent.

`<crest_factor>` is the crest factor in dB.

`<ampl_10>` is the amplitude at CCDF of 10% in dB.

`<ampl_1>` is the amplitude at CCDF of 1% in dB.

`<ampl_p1>` is the amplitude at CCDF of 0.1% in dB.

`<ampl_p01>` is the amplitude at CCDF of 0.01% in dB.

`<ampl_p001>` is the amplitude at CCDF of 0.001% in dB.

`<ampl_p0001>` is the amplitude at CCDF of 0.0001% in dB.

## Examples

`FETCh:CCDF?` might return `-33.35,35.8,9.75,3.88,7.07,8.50,9.25,9.72,9.74`, indicating (average amplitude) = -33.35 dBm, (average CCDF) = 35.8%, (crest factor) = 9.75 dB, (amplitude at CCDF of 10%) = 3.88 dB, (amplitude at CCDF of 1%) = 7.07 dB, (amplitude at CCDF of 0.1%) = 8.50 dB, (amplitude at CCDF of 0.01%) = 9.25 dB, (amplitude at CCDF of 0.001%) = 9.72 dB, and (amplitude at CCDF of 0.0001%) = 9.74 dB.

## **FETCh:CCDF:{FIRST|SECond|THIRd}:X? (Query Only)**

Returns the horizontal values of the specified trace in the CCDF measurement.

The mnemonics `FIRST`, `SECond`, and `THIRd` represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

## Conditions

Measurement views: CCDF

## Group

Fetch commands

## Syntax

```
FETCh:CCDF:{FIRSt|SECond|THIRd}:X?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the horizontal value (dB) of the CCDF graph at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:CCDF:FIRSt:X? might return #41024xxx... (1024-byte data) for the horizontal values of Trace 1.

## FETCh:CCDF:{FIRSt|SECond|THIRd}:XY? (Query Only)

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

## Conditions

Measurement views: CCDF

## Group

Fetch commands

## Syntax

```
FETCh:CCDF:{FIRSt|SECond|THIRd}:XY?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

$\langle x(n) \rangle \langle y(n) \rangle$  is the horizontal value (dB) and vertical value (%) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:CCDF:FIRST:XY?` might return #41024xxxx... (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

## **FETCh:CCDF:{FIRSt|SECond|THIRd}[:Y]? (Query Only)**

Returns the vertical values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.

## Conditions

Measurement views: CCDF

## Group

Fetch commands

## Syntax

`FETCh:CCDF:{FIRSt|SECond|THIRd}[:Y]?`

## Arguments

None

## Returns

`#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<y(n)>` is the vertical value (%) of the CCDF graph at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:CCDF:FIRST:Y?` might return #41024xxxx... (1024-byte data) for the vertical values of Trace 1.

## **FETCh:CONStE[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency of the analyzer.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Fetch commands

## Syntax

`FETCh:CONSte[:MEASview<y>]:FERRor?`

## Related Commands

[\*FETCh:EVM\[:MEASview<y>\]:FERRor?\*](#)

## Arguments

None.

## Returns

`<freq_error>::<NRf>` which is the frequency error in Hz.

## Examples

`FETCh:CONSte:MEASview2:FERRor?` might return `-10.7E+3`, which is a frequency error of -10.7 kHz for measview 2 of constellation.

## FETCh:CONSte[:MEASview<y>]:RESults? (Query only)

Returns the constellation measurement results of EVM RMS, peak, and location displayed on the bottom of the screen.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Fetch commands

## Syntax

`FETCh:CONSte[:MEASview<y>]:RESults?`

## Arguments

None

## Returns

For modulation types 2|4|8|16FSK or C4FM:

FSK\_deviation\_Avg\_Leftmost, FSK\_deviation\_Avg\_Rightmost

Where

FSK\_deviation\_Avg\_Leftmost is the average FSK deviation of the left-most symbol in Hz.

FSK\_deviation\_Avg\_Rightmost is the average FSK deviation of the right-most symbol in Hz.

For all other valid modulation types:

<EVM\_RMS>, <EVM\_peak>, <location>

Where

<EVM\_RMS>::<NRf> is the RMS EVM in percent (%).

<EVM\_peak>::<NRf> is the peak EVM in percent (%).

<location>::<NRf> is the peak location in symbol number.

The time unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

FETCh:CONStE:MEASview2:RESults? might return 2.841, 3.227, 68.000, indicating that the RMS EVM is 2.841% and the peak EVM is 3.227% at symbol #68 for measview 2 of constellation.

## FETCh:CONStE[:MEASview<y>]:TRACe? (Query only)

Returns the constellation trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Fetch commands

## Syntax

FETCh:CONStE[:MEASview<y>]:TRACe?

## Arguments

None

## Returns

#<num\_digit><num\_byte><I (1)><Q (1)><I (2)><Q (2)>...<I (n)><Q (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<I(n)> and <Q(n)> are the normalized I- and Q-coordinate values at the n<sup>th</sup> data point. 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:CONStE:MEASview2:TRACe? might return #43848xxxx... (3848-byte data) for the constellation trace data for measview 2 of constellation.

## FETCh:DDEMod[:MEASview<y>]:STABLE? (Query only)

Returns the symbol table data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Symbol table

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Symbol table)

## Group

Fetch commands

## Syntax

FETCh:DDEMod[:MEASview<y>]:STABLE?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the symbol table data at the n<sup>th</sup> data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:DDEMod:MEASview1:STABLE? might return #3512xxxx... (512-byte data) for the Symbol table measurement view1.

## FETCh:DDEMod:SYNCh:WORD:LENGth? (Query only)

Returns the length of the synch word in the symbol table.

## Conditions

Measurement views: Symbol table

## Group

Fetch commands

## Syntax

```
FETCh:DDEMod:SYNCh:WORD:LENGth?
```

## Related Commands

```
FETCh:DDEMod:SYNCh:WORD:POSition?
```

## Arguments

None

## Returns

<NR1> indicates the length of the synch word in symbols.

## Examples

FETCh:DDEMOD:SYNCH:WORD:LENGTH? might return 3, indicating the length of the synch word is three symbols.

## FETCh:DDEMod:SYNCh:WORD:POSition? (Query Only)

Returns the position of the synch word in the symbol table.

## Conditions

Measurement views: Symbol table

## Group

Fetch commands

## Syntax

```
FETCh:DDEMod:SYNCh:WORD:POSition?
```

## Related Commands

```
FETCh:DDEMod:SYNCh:WORD:LENGth?
```

## Arguments

None

## Returns

<NR1> indicates what symbol number the synch word begins at in the table. Zero (0) represents the first symbol in the table. A "-1" indicates that the synch word was not found.

## Examples

`FETCH:DDEMOD:SYNCH:WORD:POSITION?` might return 10, indicating the synch word begins at 11th symbol in the table.

## **FETCh:DIQVtime[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the Demod I&Q versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:DIQVtime[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

`FETCh:DIQVtime:MEASview1:FERRor?` might return -10.7E+3, indicating the frequency error is -10.7 kHz measurement view 1.

## **FETCh:DIQVtime[:MEASview<y>]:I? (Query only)**

Returns the I versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Fetch commands



## Syntax

```
FETCh:DIQVtime[:MEASview<y>]:I?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the I level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:DIQVtime:MEASview1:I? might return #3160xxx... (160-byte data) for the I versus Time trace measurement view 1.

## FETCh:DIQVtime[:MEASview<y>]:Q? (Query only)

Returns the Q versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:DIQVtime[:MEASview<y>]:Q?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n) > is the Q level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:DIQVtime:MEASview1:Q? might return #3160xxxx... (160-byte data) for the Q versus Time trace measurement view 1.

## FETCh:DPX:DDENsity? (Query Only)

Returns the calculated density for the selected region in the DPXogram measurement.

### Conditions

Measurement views: DPX spectrum

### Group

Fetch commands

### Syntax

FETCh:DPX:DDENsity?

### Arguments

None

### Returns

<NR1> value.

## Examples

FETCh:DPX:DDENSITY? might return a DPX signal density value of 24.4802413393.

## FETCh:DPX:DGRam:LINE:COUNT? (Query Only)

Returns the number of lines in the DPXogram measurement.

### Conditions

Measurement views: DPX spectrum

### Group

Fetch commands

### Syntax

FETCh:DPX:DGRam:LINE:COUNT?

### Arguments

None

## Returns

<NR1> number of lines in the measurement.

## Examples

`FETCH:DPX:DGRAM:LINE:COUNT?` might return 40 indicating that there were 40 lines in the measurement.

## **FETCh:DPX:DGRam:TIME[:SCALE]:OFFSet? (Query Only)**

Returns the time scale offset in the DPXogram measurement.

## Conditions

Measurement views: DPX spectrum

## Group

Fetch commands

## Syntax

`FETCh:DPX:DGRam:TIME[:SCALE]:OFFSet?`

## Arguments

None

## Returns

<string> The time offset in the DPXogram measurement in seconds.

## Examples

`FETCH:DPX:DGRAM:TIME[:SCALE]:OFFSET?` might return "0.000 s" indicating that the time scale offset was 0.000 s.

## **FETCh:DPX:RESults:TRACe<x>? (Query Only)**

Returns waveform data of specified trace <x> in the DPX spectrum measurement, where x is 1 to 5. The traces 1–4 are in the standard form. Trace 5 is the bitmap trace and its data is returned in a binary block. Trace 6 is the DPXogram trace on DPX spectrum plots. Trace 7 is the Ogram line in the DPXogram display.

## Conditions

Measurement views: DPX spectrum

## Group

Fetch commands

## Syntax

`FETCh:DPX:RESults:TRACe<x>?`

## Arguments

<NR1>

## Returns

For traces 1 to 4:

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude of the trace sample at the nth data point.

<data(n) is in the 4-byte little endian floating-point format specified in IEEE 488.2.

The format is a binary block of 32 bit floating point values. Each value ranges from 0 to 1.0, where 1.0 represents 100%.

## Examples

FETCH:DPX:RESULTS:TRACE1? might return #42004xxxx... (2004-byte of data) for the waveform data of trace one (1).

## FETCh:DPX:TDM:FREQuency:TXBer:BER? (Query Only)

Returns the Tx BER measurement BER results.

## Conditions

Measurement views: DPX:Frequency

## Group

Fetch commands

## Syntax

```
FETCh:DPX:TDM:FREQuency:TXBer:BER?
```

## Related Commands

## Arguments

None

## Returns

<NR1> is the calculated BER.

## Examples

FETCH:DPX:TDM:FREQUENCY:TXBER:BER? might return 9.1135684130E-6.

## **FETCh:DPX:TDM:FREQuency:TXBer:BITS? (Query Only)**

Returns the number of bits counted in the active Tx BER measurement session.

### **Conditions**

Measurement views: DPX:Frequency

### **Group**

Fetch commands

### **Syntax**

FETCh:DPX:TDM:FREQuency:TXBer:BITS?

### **Related Commands**

### **Arguments**

None

### **Returns**

<NR1> is the number of bits counted in the current Tx BER measurement session.

### **Examples**

FETCh:DPX:TDM:FREQUENCY:TXBER:BITS? might return 72.082000000E+12.

## **FETCh:DPX:TDM:FREQuency:TXBer:ERRors? (Query Only)**

Returns the number of errors counted in the active Tx BER measurement session.

### **Conditions**

Measurement views: DPX:Frequency

### **Group**

Fetch commands

### **Syntax**

FETCh:DPX:TDM:FREQuency:TXBer:ERRors?

### **Arguments**

None

### **Returns**

<NR1> is the number of errors counted in the current Tx BER measurement session.

## Examples

`FETCH:DPX:TDM:FREQUENCY:TXBER:ERRORS?` might return 1.0000000000.

## **FETCh:DPX:TDM:FREQuency:TXBer:STATe? (Query Only)**

Returns the state of the Tx BER measurement.

### Conditions

Measurement views: DPX:Frequency

### Group

Fetch commands

### Syntax

`FETCh:DPX:TDM:FREQuency:TXBer:STATe?`

### Returns

`IDLE` indicates the Tx BER measurement is not running.

`ACTIVE` indicates the Tx BER measurement is in progress.

`SYNC` indicates the instrument is in the process of syncing with the data pattern.

`DONE` indicates the Tx BER measurement is finished.

## Examples

`FETCH:DPX:TDM:FREQUENCY:TXBER:STATE?` might return `ACTIVE` indicating the instrument is measuring the Tx BER.

## **FETCh:DPX:TRACe:AVERAge? (Query Only)**

Returns waveform data of the average trace in the DPX spectrum measurement.

### Conditions

Measurement views: DPX spectrum

### Group

Fetch commands

### Syntax

`FETCh:DPX:TRACe:AVERAge?`

### Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude of the average trace at the nth data point in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

FETCH:DPX:TRACE:AVERAGE? might return #42004xxxx... (2004-byte data) for the waveform data of the average trace.

## FETCh:DPX:TRACe:BITMap? (Query Only)

Returns a data block of the DPX spectrum measurement bitmap waveform data.

## Conditions

Measurement views: DPX spectrum

## Group

Fetch commands

## Syntax

```
FETCh:DPX:TRACe:BITMap?
```

## Arguments

None

## Returns

A binary block of 32-bit floating point values (IEEE standard single precision floating point format). Each value ranges from 0 to 1.0, where 1.0 represents 100%. Bitmaps have 201 rows and 801 columns. Data[0] is the first column in the first row, data[801] is the first column in the 2nd row, and so on.

Bitmap values are returned in the following order: Top left pixel first, then the rest of the top row, after that row by row. In other words, the pixels in the bitmap are returned from left to right and top to bottom; one row at a time.

## Examples

FETCH:DPX:TRACE:BITMAP? might return #6644004xxxx...

## FETCh:DPX:TRACe:MATH? (Query Only)

Returns waveform data of the math trace in the DPX spectrum measurement.

## Conditions

Measurement views: DPX spectrum

## Group

Fetch commands

## Syntax

```
FETCh:DPX:TRACe:MATH?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude of the math trace at the nth data point in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:DPX:TRACe:MATH? might return #42004xxxx... (2004-byte data) for the waveform data of the math trace.

## FETCh:DPX:TRACe:MAXimum? (Query Only)

Returns waveform data of the maximum trace in the DPX spectrum measurement.

## Conditions

Measurement views: DPX spectrum

## Group

Fetch commands

## Syntax

```
FETCh:DPX:TRACe:MAXimum?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```



Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude of the maximum trace at the nth data point in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSE]:POWER:UNITS command.

## Examples

FETCH:DPX:TRACE:MAXIMUM? might return #42004xxxx... (2004-byte data) for the waveform data of the maximum trace.

## FETCh:DPX:TRACe:MINimum? (Query Only)

Returns waveform data of the minimum trace in the DPX spectrum measurement.

### Conditions

Measurement views: DPX spectrum

### Group

Fetch commands

### Syntax

FETCh:DPX:TRACe:MINimum?

### Arguments

None

### Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude data of the minimum trace at the nth data point in dBm,

4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSE]:POWER:UNITS command.

## Examples

FETCH:DPX:TRACE:MINIMUM? might return #42004xxxx... (2004-byte data) for the waveform data of the minimum trace.

## FETCh:EDlagram[:MEASview<y>]:FDEVIation? (Query only)

Returns the frequency deviation versus Time trace data with the X values.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Fetch commands

## Syntax

```
FETCh:EDIagram[:MEASview<y>]:FDEVIation?
```

## Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> is the frequency deviation in Hz and time (symbols) pair at the n<sup>th</sup> data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:EDIagram:MEASview2:FDEVIation? might return #3160xxxx... (160-byte data) for the frequency deviation versus Time trace for measview2 of the Eye diagram.

## FETCh:EDIagram[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the eye diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Fetch commands

## Syntax

```
FETCh:EDIagram[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`FETCh:EDIagram:MEASview2:FERRor?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz for measview2 of the Eye diagram.

## FETCh:EDIagram[:MEASview<y>]:I? (Query only)

Returns the I versus Time trace data in the eye diagram measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Fetch commands

## Syntax

```
FETCh:EDIagram[:MEASview<y>]:I?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<Y(n)><X(n)>` is the I level (normalized) and time (symbols) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:EDIagram:MEASview2:I?` might return `#3160xxxx...` (160-byte data) for the I versus Time trace for measview2 of the Eye diagram.

## **FETCh:EDIagram[:MEASview<y>]:Q? (Query only)**

Returns the Q versus Time trace data in the eye diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

### **Group**

Fetch commands

### **Syntax**

```
FETCh:EDIagram[:MEASview<y>]:Q?
```

### **Arguments**

None

### **Returns**

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)><X(n)> is the Q level (normalized) and time (symbols) pair at the n<sup>th</sup> data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### **Examples**

FETCh:EDIagram:MEASview2:Q? might return #3160xxx... (160-byte data) for the Q versus Time trace for measview2 of the Eye digram.

## **FETCh:EMC:COUNT? (Query only)**

Queries the total numbers of spots detected in EMC View.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Fetch commands

## Syntax

```
FETCh:EMC:COUNT?
```

## Arguments

None

## Returns

<NR1>

## Examples

FETCH:EMC:COUNT? might return 15, indicating that 15 spots are detected.

## FETCh:EMC:EMISsion:RESUltS:STATUs? (Query Only)

Queries the status of the emission results in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:EMISsion:RESUltS:STATUs?
```

## Arguments

None

## Returns

<String> Either PASS or FAIL

## Examples

FETCH:EMC:EMISSION:RESULTS:STATUS? returns PASS, indicating that the status of emission results is passed.

## FETCh:EMC:SPOT<x>:AMBlent:AMPLitude? (Query Only)

Queries the amplitude of the selected spot in ambient trace of EMC view.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:AMBIent:AMPLitude?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT1:AMBIent:AMPLitude?` might return 52.612, indicating that the amplitude of spot 1 in the ambient trace is 52.61 dBuV.

## FETCh:EMC:SPOT<x>:AMBIent:DELTA? (Query Only)

Queries the Delta value of the selected spot in ambient trace of EMC View.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:AMBIent:DELTA?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT2:AMBIent:DELTA?` might return 3.3614, indicating that the delta value of spot 2 in the ambient trace is 3.36 dBuV.

## FETCh:EMC:SPOT<x>:AMBIent:PASS? (Query Only)

Queries the status of the selected spot in ambient trace in EMC view. The default status is THRE.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:AMBIent:PASS?
```

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

FETCh:EMC:SPOT3:AMBIent:PASS? might return THRE, indicating that spot 3 of the ambient trace is failing by crossing the threshold.

## FETCh:EMC:SPOT<x>:FREQuency? (Query Only)

Queries the frequency of the specified spot in EMC View.

<x> = <NR1> indicates the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:FREQuency?
```

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT2:FREQuency?` might return `366.1250000E+3`, indicating that the frequency of spot 2 is 366.125 kHz.

## FETCh:EMC:SPOT<x>:PASS? (Query Only)

Queries the overall status of the specified spot in EMC view. The default status is THRE.

<x> = <NR1> indicates the spot number.

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:PASS?`

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

`FETCh:EMC:SPOT1:PASS?` might return OLIM, indicating that spot 1 is failing by crossing over the limit.

## FETCh:EMC:SPOT<x>:RANGe? (Query Only)

Queries the range of the specified spot in EMC View.

<x> = <NR1> indicates the spot number.

## Conditions

Measurement view: EMC - EMI Display



## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:RANGe?`

## Arguments

None

## Returns

<String> from A to T

## Examples

`FETCh:EMC:SPOT1:RANGe?` might return "A", indicating that the range of spot 1 is "A".

## FETCh:EMC:SPOT<x>:RECAll:AMPLitude? (Query Only)

Queries the amplitude of the selected spot in recalled trace of EMC View.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:RECAll:AMPLitude?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT1:RECAll:AMPLitude?` might return 23.598, indicating that the amplitude of spot 1 in the recalled trace is 23.59 dBuV.

## FETCh:EMC:SPOT<x>:RECAll:DELTA? (Query Only)

Queries the Delta value of the specified spot in recalled trace of EMC View.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:RECAll:DELTA?
```

## Arguments

None

## Returns

<NRf>

## Examples

FETCh:EMC:SPOT2:RECAll:DELTA? might return 5.5878, indicating that delta value of the spot 2 in the recalled trace is 5.58 dBuV.

## FETCh:EMC:SPOT<x>:RECAll:PASS? (Query Only)

Queries the status of the selected spot in recalled trace in EMC view. The default status is THREE.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:RECAll:PASS?
```

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

`FETCh:EMC:SPOT1:RECALL:PASS?` might return OMAR, indicating that spot 1 of the recalled trace is failing by crossing the Margin.

## **FETCh:EMC:SPOT<x>:SCDEtector<y>:AMPLitude? (Query Only)**

Queries the amplitude of the specified spot with respect to specified scan detector in the results table.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:SCDEtector<y>:AMPLitude?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT1:SCDEtector1:AMPLitude?` might return 20.6209, indicating that the amplitude of spot 1 for scan detection 1 is 20.62 dBuV.

## **FETCh:EMC:SPOT<x>:SCDEtector<y>:DELTA? (Query Only)**

Queries the Delta value of the specified spot with respect to specified scan detector in the results table.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:SCDEtector<y>:DELTA?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT2:SCDEtector1:DELTA?` might return 3.6209, indicating that the delta value of spot 2 for scan detection 1 is 3.62 dBuV.

## **FETCh:EMC:SPOT<x>:SCDEtector<y>:PASS? (Query Only)**

Queries the status of the specified spot based on the specified scan detector in EMC view. The default status is THRE.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:SCDEtector<y>:PASS?`

## Arguments

None

## Returns

<String> either OLIM, OMAR or THRE

where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

`FETCh:EMC:SPOT2:SCDEtector1:PASS?` might return THRE, indicating that spot 2 of scan detector 1 is failing by crossing the threshold.

## **FETCh:EMC:SPOT<x>:SPDEtector<y>:AMPLitude? (Query Only)**

Queries the amplitude of the specified spot with respect to specified spot detector in the results table.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:SPDEtector<y>:AMPLitude?
```

## Arguments

None

## Returns

<NRf>

## Examples

FETCh:EMC:SPOT1:SPDEtector1:AMPLitude? might return 18.365, indicating that amplitude of spot 1 for spot detection 1 is 18.36 dBuV.

## FETCh:EMC:SPOT<x>:SPDEtector<y>:DELTA? (Query Only)

Queries the Delta value of the specified spot with respect to specified spot detector in the results table.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

```
FETCh:EMC:SPOT<x>:SPDEtector<y>:DELTA?
```

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:EMC:SPOT2:SPDEtector1:DELta?` might return 2.2578, indicating that delta value of spot 2 for spot detection 1 is 2.25 dBuV.

## **FETCh:EMC:SPOT<x>:SPDEtector<y>:PASS? (Query Only)**

Queries the status of the selected spot based on the specified spot detector in EMC view. The default status is THRE.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Fetch commands

## Syntax

`FETCh:EMC:SPOT<x>:SPDEtector<y>:PASS??`

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

`FETCh:EMC:SPOT1:SPDEtector2:PASS?` might return OMAR, indicating that the spot 1 of spot detector 2 is failing by crossing Margin.

## **FETCh:EVM[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:EVM[:MEASVIEW<y>]:FERRor?
```

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

FETCh:EVM:MEASview2:FERRor? might return -10.7E+3, indicating the frequency error is -10.7 kHz for measview2 of EVM.

## FETCh:EVM[:MEASview<y>]:PEAK? (Query only)

Returns the peak value in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:EVM[:MEASview<y>]:PEAK?
```

## Related Commands

[FETCh:EVM\[:MEASVIEW<y>\]:PINdex?](#)

## Arguments

None

## Returns

<peak>::<NRf> is the peak EVM value in percent (%).

## Examples

`FETCh:EVM:MEASview2:PEAK?` might return 1.32, indicating the peak EVM value is 1.32% for measview2 of EVM.

## FETCh:EVM[:MEASview<y>]:PINdex? (Query only)

Returns the time at the EVM peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Fetch commands

## Syntax

`FETCh:EVM[:MEASview<y>]:PINdex?`

## Related Commands

[FETCh:EVM\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

`<peak_time>::<Nrf>` is the time at the EVM peak in symbol number.

The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`FETCh:EVM:MEASview2:PINdex` might return 68.000, indicating that the EVM peak is at symbol #68 for measview2 of EVM.

## FETCh:EVM[:MEASview<y>]:RMS? (Query only)

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).



## Group

Fetch commands

## Syntax

```
FETCh:EVM[:MEASview<y>]:RMS?
```

## Arguments

None

## Returns

<rms>::<NRf> is the RMS EVM value in percent (%).

## Examples

FETCh:EVM:MEASview2:RMS? might return 0.582, indicating the RMS EVM value is 0.582% for the measview2 of EVM.

## FETCh:EVM[:MEASview<y>]:TRACe? (Query only)

Returns the EVM versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 for channels MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:EVM[:MEASview<y>]:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the EVM value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:EVM:MEASview2:TRACe?` might return `#42036xxx...` (2036-byte data) for the EVM versus Time Measview 2.

## **FETCh:FDVTime[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the Frequency deviation versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

### Group

Fetch commands

### Syntax

`FETCh:FDVTime:MEASview<y>:FERRor?`

### Arguments

None

### Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

### Examples

`FETCh:FDVTime:MEASview2:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

## **FETCh:FDVTime[:MEASview<y>]:TRACe? (Query only)**

Returns the Frequency deviation versus Time trace data.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

### Group

Fetch commands

## Syntax

```
FETCh:FDVTime:MEASview<y>:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the frequency deviation value at the n<sup>th</sup> data point in percent (Hz), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:FDVTime:MEASview2:TRACe? might return #3160xxx... (160-byte data) for the Frequency deviation versus Time Measview 2.

## FETCh:{FM|PM}:FERRor? (Query Only)

Returns the frequency error in the Frequency modulation or Phase modulation measurements.

## Conditions

Measurement views: Frequency deviation versus Time

## Group

Fetch commands

## Syntax

```
FETCh:{FM|PM}:FERRor?
```

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

FETCh:{FM|PM}:FERROR? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

## FETCh:FM:PHALf? (Query Only)

Returns the half peak-peak frequency deviation (Pk-Pk/2) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

FETCh:FM:PHALf?

## Arguments

None

## Returns

$\langle Pk-Pk/2 \rangle : : \langle NRf \rangle$  is the half peak-peak frequency deviation in Hz.

## Examples

FETCh:FM:PHALF? might return 628.9E+3, indicating the half peak-peak frequency deviation is 628.9 kHz.

## FETCh:FM:PNEGative? (Query Only)

Returns the negative peak frequency deviation ( $-Pk$ ) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

FETCh:FM:PNEGative?

## Arguments

None

## Returns

$\langle -Pk \rangle : : \langle NRf \rangle$  is the negative peak frequency deviation in Hz.

## Examples

FETCh:FM:PNEGATIVE? might return  $-495.6E+3$ , indicating the negative peak frequency deviation is  $-495.6$  kHz.

## FETCh:FM:PPOSitive? (Query Only)

Returns the positive peak frequency deviation ( $+Pk$ ) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

`FETCh:FM:PPOSitive?`

## Arguments

None

## Returns

`<+Pk> : : <NRf>` is the positive peak frequency deviation in Hz.

## Examples

`FETCH:FM:PPOSITIVE?` might return `763.2E+3`, indicating the positive peak frequency deviation is 763.2 kHz.

## FETCh:FM:PTPeak? (Query Only)

Returns the peak-peak frequency deviation (Pk-Pk) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

`FETCh:FM:PTPeak?`

## Arguments

None

## Returns

`<Pk-Pk> : : <NRf>` is the peak-peak frequency deviation in Hz.

## Examples

`FETCH:FM:PTPEAK?` might return `1.258E+6`, indicating the peak-peak frequency deviation is 1.258 MHz.

## FETCh:FM:RESult? (Query Only)

Returns the FM measurement results.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

`FETCh:FM:RESult?`

## Arguments

None

## Returns

`<+Pk>, <-Pk>, <RMS>, <Pk-Pk>, <Pk-Pk/2>`

Where

`<+Pk> : : <NRf>` is the positive peak frequency deviation in Hz.

`<-Pk> : : <NRf>` is the negative peak frequency deviation in Hz.

`<RMS> : : <NRf>` is the RMS frequency deviation in Hz.

`<Pk-Pk> : : <NRf>` is the peak-peak frequency deviation in Hz.

`<Pk-Pk/2> : : <NRf>` is the half peak-peak frequency deviation in Hz.

## Examples

`FETCH:FM:RESULT?` might return `763.2E+3,-494.6E+3,271.2E+3,1.258E+6,628.9E+3`.

## FETCh:FM:RMS? (Query Only)

Returns the RMS frequency deviation in the FM measurement.

## Conditions

Measurement views: FM

## Group

Fetch commands

## Syntax

`FETCh:FM:RMS?`

## Arguments

None

## Returns

`<RMS> : : <NRf>` is the RMS frequency deviation in Hz.

## Examples

`FETCH:FM:RMS?` might return `271.2E+3`, indicating the RMS frequency deviation is 271.2 kHz.

## **FETCh:{FSETtling|PSETtling}:FTTime? (Query Only)**

Returns the settling time from the trigger position in seconds. The settling time is measured from the trigger point (see `:FETCh:FSETtling:TRIGger:TIME`).

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Fetch commands

### Syntax

`FETCh:{FSETtling|PSETtling}:FTTime?`

### Arguments

None

### Returns

`<value>::<NRf>` is the settling time in seconds.

## Examples

`FETCH:FSETTLING:FTTIME?` might return `44.8300E-6`, indicating the settling time is 44.83  $\mu$ s.

## **FETCh:{FSETtling|PSETtling}:MASK[:PASS]? (Query Only)**

Returns whether the input signal passes the mask test.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Fetch commands

### Syntax

`FETCh:{FSETtling|PSETtling}:MASK[:PASS]?`

### Arguments

None

## Returns

0 is returned if the signal fails the mask test.

1 is returned if the signal passes the mask test.

## Examples

`FETCH:FSETTLING:MASK:PASS?` might return 1, indicating the signal passed (did not exceed the mask limits).

## FETCh:{FSETtling|PSETtling}:SETTled:FREQuency? (Query Only)

Returns the frequency at which the signal is considered settled.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:SETTled:FREQuency?`

## Arguments

None

## Returns

`<value>::<NRf>` is the frequency at which the signal is settled.

## Examples

`FETCH:FSETTLING:SETTLED:FREQUENCY?` might return 2.44838E+9, indicating the input signal frequency at the point where the signal is considered settled is 2.44838 GHz.

## FETCh:{FSETtling|PSETtling}:SETTled[:PASS]? (Query Only)

Returns whether the input signal is settled with the tolerance range.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:SETTled[:PASS]?`



## Arguments

None

## Returns

0 is returned if the signal is not settled within the tolerance range.

1 is returned if the signal is settled within the tolerance range.

## Examples

`FETCH:FSETTLING:SETTLED:PASS?` might return 1, indicating the signal is settled within the tolerance range.

## **FETCh:{FSETtling|PSETtling}:SETTled:TIME? (Query Only)**

Returns the settled time in seconds. The settled time is measured from the measurement start point.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:SETTled:TIME?`

## Arguments

None

## Returns

`<value>::<NRf>` is the settled time in seconds.

## Examples

`FETCH:FSETTLING:SETTLED:TIME?` might return 299.830000E-6, indicating the settled time is 299.83  $\mu$ s.

## **FETCh:{FSETtling|PSETtling}:SLMSd[:PASS]? (Query Only)**

Returns whether the input signal is settled within the specified tolerance and the signal is settled longer than the Minimum Settled Duration (`[SENSe]:{FSETtling|PSETtling}:SDURation:MINimum`).

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:SLMSd[:PASS]?`

## Arguments

None

## Returns

0 is returned if the signal is not settled within the tolerance range and minimum settled duration.

1 is returned if the signal is settled within the tolerance range and minimum settled duration.

## Examples

`FETCH:FSETTLING:SLMSD:PASS?` might return 1, indicating the signal is settled within the tolerance range and the minimum settled duration.

## **FETCh:{FSETtling|PSETtling}:START:TIME? (Query Only)**

Returns the start time for the measurement in seconds. The start time is measured from the start of the analysis period.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:START:TIME?`

## Arguments

None

## Returns

`<value>::<NRf>` is the time in seconds when the measurement started.

## Examples

`FETCH:FSETTLING:START:TIME?` might return `251.4300E-6`, indicating the measurement started 251.43  $\mu$ s after the beginning of the analysis period.

## **FETCh:{FSETtling|PSETtling}:TIME? (Query Only)**

Returns the settling time in seconds. The settling time is measured from the start time (see `:FETCh:FSETtling:START:TIME`).

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

```
FETCh:{FSETtling|PSETtling}:TIME?
```

## Arguments

None

## Returns

<value>: : <Nrf> is the settling time in seconds.

## Examples

FETCH:FSETTLING:TIME? might return 48.4000E-6, indicating the settling time is 48.4  $\mu$ s.

## FETCh:{FSETtling|PSETtling}:TRACe<x>:X? (Query Only)

Returns the Frequency or Phase values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

```
FETCh:{FSETtling|PSETtling}:TRACe<x>:X?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><y(1)><y(2)>...<y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the frequency (Hz) or phase (degrees) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:FSETTLING:TRACE1:X? might return #574232xxxx... (74232-byte data) for the frequency values of Trace 1.

## FETCh:{FSETtling|PSETtling}:TRACe<x>:XY? (Query Only)

Returns the time and frequency or phase value pairs of the specified trace. The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Fetch commands

### Syntax

```
FETCh:{FSETtling|PSETtling}:TRACe<x>:XY?
```

### Arguments

None

### Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the time and frequency or phase value pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

FETCh:FSETTLING:TRACE2:XY? might return #574232xxxx... (74232-byte data) for the time and frequency pairs of the Trace 2.

## FETCh:{FSETtling|PSETtling}:TRACe<x>[:Y]? (Query Only)

Returns the frequency or phase values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Fetch commands

### Syntax

```
FETCh:{FSETtling|PSETtling}:TRACe<x>[:Y]?
```

## Arguments

None

## Returns

#<num\_digit><num\_byte><y(1)><y(2)>...<y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the frequency or phase value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:FSETTLING:TRACE1:Y? might return #575148xxxx... (75148-byte data) for the frequency values of Trace 1.

## FETCh:{FSETtling|PSETtling}:TRIGger:TIME? (Query Only)

Returns the time when the trigger occurred in seconds.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Fetch commands

### Syntax

FETCh:{FSETtling|PSETtling}:TRIGger:TIME?

### Arguments

None

### Returns

<value> : :<NRf> is the time in seconds after the measurement start point when the trigger occurred.

### Examples

FETCH:FSETTLING:TRIGGER:TIME? might return 2.255E-6, indicating the time when the trigger occurred 255  $\mu$ s after the start of the measurement.

## FETCh:{FSETtling|PSETtling}:VALue? (Query Only)

Returns the settled value in Hz for Frequency Settling and in degrees for Phase Settling.

### Conditions

Measurement views: Frequency and Phase Settling

## Group

Fetch commands

## Syntax

`FETCh:{FSETtling|PSETtling}:VALue?`

## Arguments

None

## Returns

`<value>::<NRf>` is the settling value in Hz for Frequency Settling and in degrees for Phase Settling.

## Examples

`FETCH:FSETTLING:VALUE?` might return `2.44838155E+9`, indicating the settled frequency is 2.44838 GHz.

## FETCh:FSETtling:ERRor? (Query Only)

Returns the settled error in Hz in Frequency Settling. Only for Frequency Settling Time measurement. In Frequency Settling:

- When Target Reference is set to Auto, Settled Error = 0.

## Conditions

Measurement views: Frequency Settling

## Group

Fetch commands

## Syntax

`FETCh:FSETtling:ERRor?`

## Arguments

None

## Returns

`<value>::<NRf>` returns the settled error in Hz.

When Target Reference is set to Auto, Settled Error = 0.

When Target Reference is set to Meas Freq:

Settled Error = Settled Frequency – (Measurement Frequency + Offset)

## Examples

`FETCH:FSETTLING:FTTIME?` might return 0, indicating the Target Reference is set to Auto.

## FETCh:FVTime[:MEASview<y>]? (Query Only)

Returns the Frequency versus Time trace data.

### Conditions

Measurement views: Frequency versus Time

### Group

Fetch commands

### Syntax

```
FETCh:FVTime[:MEASview<y>]?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the frequency in Hz at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

FETCh:FVTIME:MEASview1? might return #3156xxxx... (156-byte data) for the Frequency versus Time trace.

## FETCh:FVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Fetch commands

### Syntax

```
FETCh:FVTime[:MEASview<y>]:MAXimum?
```

### Related Commands

```
FETCh:FVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max> : : <NRf> is the maximum frequency drift in Hz.

## Examples

FETCH:FVTIME:MEASview1:MAXIMUM? might return 2.625E+6, indicating the maximum frequency drift is 2.625 MHz.

## FETCh:FVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the frequency drift is maximum.

## Conditions

Measurement views: Frequency versus Time

## Group

Fetch commands

## Syntax

FETCh:FVTime[:MEASview<y>]:MAXLocation?

## Related Commands

FETCh:FVTime[:MEASview<y>]:MAXimum?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max\_time>::<NRf> is the time in seconds at which the frequency drift is maximum.

## Examples

FETCH:FVTIME:MEASview1:MAXLOCATION? might return 25.03E-9, indicating the frequency drift is maximum at 25.03 ns.

## FETCh:FVTime[:MEASview<y>]:MINimum? (Query Only)

Returns the minimum value in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time



## Group

Fetch commands

## Syntax

```
FETCh:FVTime[:MEASview<y>]:MINimum?
```

## Related Commands

```
FETCh:FVTime[:MEASview<y>]:MINLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min>::<NRf> is the minimum frequency drift in Hz.

## Examples

FETCH:FVTIME:MEASview1:MINIMUM? might return -6.618E+6, indicating the minimum frequency drift is -6.618 MHz.

## FETCh:FVTime[:MEASview<y>]:MINLocation? (Query Only)

Returns the time at which the frequency drift is minimum.

## Conditions

Measurement views: Frequency versus Time

## Group

Fetch commands

## Syntax

```
FETCh:FVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
FETCh:FVTime[:MEASview<y>]:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<NRf> is the time in seconds at which the frequency drift is minimum.

## Examples

`FETCH:FVTIME:MEASview1:MINLOCATION?` might return 450.7E-9, indicating the frequency drift is minimum at 450.7 ns.

## FETCh:FVTime[:MEASview<y>]:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

### Conditions

Measurement views: Frequency versus Time

### Group

Fetch commands

### Syntax

`FETCh:FVTime[:MEASview<y>]:RESult?`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

<max>, <max\_time>, <min>, <min\_time>

Where

<max>::<NRf> is the maximum frequency drift in Hz.

<max\_time>::<NRf> is the time in seconds at which the frequency drift is maximum.

<min>::<NRf> is the minimum frequency drift in Hz.

<min\_time>::<NRf> is the time in seconds at which the frequency drift is minimum.

## Examples

`FETCH:FVTIME:MEASview1:RESULT?` might return 2.625E+6,25.03E-9,-6.618E+6,450.7E-9, indicating the maximum frequency drift is 2.625 MHz at 25.03 ns and the minimum frequency drift is -6.618 MHz at 450.7 ns.

## FETCh:IQVTime[:MEASview<y>]:I? (Query Only)

Returns the I versus Time trace data.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Fetch commands

## Syntax

```
FETCh:IQVTime[:MEASview<y>]:I?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the I level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:IQVTIME:MEASview1:I? might return #3160xxxx... (160-byte data) for the I versus Time trace.

## FETCh:IQVTime[:MEASview<y>]:MAXimum (Query Only)

Returns the maximum value in the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Fetch commands

## Syntax

```
FETCh:IQVTime[:MEASview<y>]:MAXimum
```

## Related Commands

```
FETCh:IQVTime[:MEASview<y>]:MAXLocation
```

## Arguments

None

## Returns

<max>::<NRf> is the maximum I or Q level in volts.

Use the TRACe:IQVTime[:MEASview<y>]:SElect:I or TRACe:IQVTime[:MEASview<y>]:SElect:Q command to select the trace.

## Examples

`FETCh:IQVTime:MEASview1:MAXimum?` might return 1.214, indicating the maximum I or Q level is 1.214 V.

## FETCh:IQVTime[:MEASview<y>]:MAXLocation (Query Only)

Returns the time at which the I or Q level is maximum.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Fetch commands

### Syntax

```
FETCh:IQVTime[:MEASview<y>]:MAXLocation
```

### Related Commands

```
FETCh:IQVTime[:MEASview<y>]:MAXimum
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

<max\_time>::<NRf> is the time in seconds at which the I or Q level is maximum.

## Examples

`FETCh:IQVTime:MEASview1:MAXLocation` might return 175.3E-9, indicating the I or Q level is maximum at 175.3 ns.

## FETCh:IQVTime[:MEASview<y>]:MINimum (Query Only)

Returns the minimum value in the RF I&Q versus Time measurement.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Fetch commands

### Syntax

```
FETCh:IQVTime[:MEASview<y>]:MINimum
```

## Related Commands

`FETCh:IQVTime[:MEASview<y>]:MINILocation`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min>::<NRf> is the minimum I or Q level in volts.

Use the `TRACe:IQVTime[:MEASview<y>]:SElect:I` or `TRACe:IQVTime[:MEASview<y>]:SElect:Q` command to select the trace.

## Examples

`FETCh:IQVTime:MEASview1:MINimum?` might return `-370.5E-3`, indicating the minimum I or Q level is `-370.5 mV`.

## FETCh:IQVTime[:MEASview<y>]:MINLocation (Query Only)

Returns the time at which the I or Q level is minimum.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Fetch commands

## Syntax

`FETCh:IQVTime[:MEASview<y>]:MINLocation`

## Related Commands

`FETCh:IQVTime[:MEASview<y>]:MINimum`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<NRf> is the time in seconds at which the I or Q level is minimum.

## Examples

`FETCh:IQVTime:MEASview1:MINLocation` might return `450.7E-9`, indicating the I or Q level is minimum at `450.7 ns`.

## FETCh:IQVTime[:MEASview<y>]:Q (Query Only)

Returns the Q versus Time trace data.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Fetch commands

## Syntax

```
FETCh:IQVTime[:MEASview<y>]:Q
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the Q level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:IQVTime:MEASview1:Q? might return #3160xxx... (160-byte data) for the Q versus Time trace.

## FETCh:IQVTime[:MEASview<y>]:RESult (Query Only)

Returns the RF I&Q versus Time measurement results.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Fetch commands

## Syntax

```
FETCh:IQVTime[:MEASview<y>]:RESult
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
<max>,<max_time>,<min>,<min_time>
```

Where

<max>::<NRf> is the maximum I or Q level in volts.

<max\_time>::<NRf> is the time in seconds at which the I or Q level is maximum.

<min>::<NRf> is the minimum I or Q level in volts.

<min\_time>::<NRf> is the time in seconds at which the I or Q level is minimum.

Use the `TRACe:IQVTime[:MEASview<y>]:SElect:I` or `TRACe:IQVTime:MEASview<y>:SElect:Q` command to select the trace.

## Examples

`FETCh:IQVTime:MEASview1:RESult?` might return `1.214,175.3E-9,-370.5E-3,450.7E-9`, indicating the maximum I or Q level is 1.214 V at 175.3 ns and the minimum I or Q level is -370.5 mV at 450.7 ns.

## FETCh:LTE:ACLR:MHITs:COUNT? (Query Only)

Returns the number of rows in the results table of the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Fetch commands

### Syntax

`FETCh:LTE:ACLR:MHITs:COUNT?`

### Returns

<NR1>

### Examples

`FETCh:LTE:ACLR:MHITs:COUNT?` might return 7, indicating there are seven rows in the results table.

## FETCh:LTE:ACLR:MHITs<x>:BANDwidth? (Query Only)

Returns the bandwidth value of the channel (row) of the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

### Group

Fetch commands

## Syntax

`FETCh:LTE:ACLR:MHITs<x>:BANDwidth?`

## Returns

<NR3>

## Examples

`FETCh:LTE:ACLR:MHITS1:BANDWIDTH?` might return `4.515000000E+6`, which indicates a bandwidth for Channel 1 of 4.515 MHz.

## **FETCh:LTE:ACLR:MHITs<x>:CHANnel:NAME? (Query Only)**

Returns the channel name of the specified index in the results table of the LTE ACLR display. The minimum index will be 1 and the maximum index will be the count of the channels in the results table.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Fetch commands

## Syntax

`FETCh:LTE:ACLR:MHITs<x>:CHANnel:NAME?`

## Returns

<string>

## Examples

`FETCh:LTE:ACLR:MHITS2:CHANNEL:NAME?` might return "B", which is the name of channel 2.

## **FETCh:LTE:ACLR:MHITs<x>:FREQuency:OFFSet? (Query Only)**

Returns the frequency offset of the specified channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Fetch commands



## Syntax

```
FETCh:LTE:ACLR:MHITs<x>:FREQUency:OFFSet?
```

## Returns

<NR3>

## Examples

FETCh:LTE:ACLR:MHITS1:FREQUENCY:OFFSET? might return 5.000000000E+6 which indicates the frequency offset of Channel 1 is 5 MHz.

## FETCh:LTE:ACLR:MHITs<x>:INTEg:LOWEr:ABSolute? (Query Only)

Returns the lower absolute value of the channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Fetch commands

## Syntax

```
FETCh:LTE:ACLR:MHITs<x>:INTEg:LOWEr:ABSolute?
```

## Returns

<NRf>

## Examples

FETCh:LTE:ACLR:MHITS1:INTEG:LOWER:ABSOLUTE? might return -53.3921980303, which indicates the lower absolute value of Channel 1 is -53.3921980303 dBm.



**Note:** The actual units depends on the choice made from the units tab of the Analysis control panel.

## FETCh:LTE:ACLR:MHITs<x>:INTEg:LOWEr:RELative? (Query Only)

Returns the lower relative value of the channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Fetch commands

## Syntax

`FETCh:LTE:ACLR:MHITs<x>:INTEg:LOWEr:RELative?`

## Returns

<NRf>

## Examples

`FETCh:LTE:ACLR:MHITS1:INTEG:LOWER:RELATIVE?` might return -53.3921980303 which indicates lower relative value of Channel 1 is -53.3921980303 dB.

## **FETCh:LTE:ACLR:MHITs<x>:INTEg:UPPEr:ABSolute? (Query Only)**

Returns the upper absolute value of the channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Fetch commands

## Syntax

`FETCh:LTE:ACLR:MHITs<x>:INTEg:UPPEr:ABSolute?`

## Returns

<NRf>

## Examples

`FETCh:LTE:ACLR:MHITS1:INTEG:UPPER:ABSOLUTE?` might return -53.3921980303 which indicates upper absolute value of Channel 1 is -53.3921980303 dBm.

## **FETCh:LTE:ACLR:MHITs<x>:INTEg:UPPEr:RELative? (Query Only)**

Returns the upper relative value of the channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Fetch commands

## Syntax

```
FETCh:LTE:ACLR:MHITs<x>:INTEg:UPPEr:RELative?
```

## Returns

<NRf>

## Examples

READ:LTE:ACLR:MHITS1:INTEG:UPPER:RELATIVE? might return -53.3921980303 which indicates upper relative value of Channel 1 is -53.3921980303 dB.

## FETCh:LTE:ACLR:REFerence:POWer? (Query Only)

Returns the reference power level in the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Fetch commands

## Syntax

```
FETCh:LTE:ACLR:REFerence:POWer?
```

## Returns

<NRf>

## Examples

FETCH:LTE:ACLR:REFERENCE:POWER? might return -16.92, indicating that the reference power level is -16.92 dBm.

## FETCh:LTE:ACLR:RESUlts:STATUs? (Query Only)

Returns the pass or fail status of the LTE ACLR display measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Fetch commands

## Syntax

```
FETCh:LTE:ACLR:RESUlts:STATUs?
```

## Returns

0: means measurement status is FAIL.

1: means measurement status is PASS.

## Examples

`FETCH:LTE:ACLR:RESULTS:STATUS?` might return 1, indicating that the LTE ACLR measurement status is PASS.

## FETCh:LTE:ACLR:SPECtrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Fetch commands

### Syntax

`FETCh:LTE:ACLR:SPECtrum:X?`

### Returns

`#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where,

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<x(n)>` is the frequency (Hz) at the  $n^{\text{th}}$  point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

`FETCH:LTE:ACLR:SPECTRUM:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of the spectrum trace.

## FETCh:LTE:ACLR:SPECtrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Fetch commands

### Syntax

`FETCh:LTE:ACLR:SPECtrum:XY?`

### Returns

`#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>`

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the frequency (Hz) and amplitude (dBm) pair at the *n*th point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:LTE:ACLR:SPECTRUM:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and amplitude pairs of the spectrum trace.

## FETCh:LTE:ACLR:SPECtrum:Y? (Query Only)

Returns the amplitude of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Fetch commands

### Syntax

`FETCh:LTE:ACLR:SPECtrum:Y?`

### Returns

`#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the amplitude (dB) at the *n*th point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:LTE:ACLR:SPECTRUM:Y?` might return `#516020xxxx...` (16020-byte data) for the amplitude of the spectrum trace.

## FETCh:LTE:CHSPectrum:OBW? (Query Only)

Returns the Occupied Bandwidth (OBW) in Hz of the LTE Channel Spectrum measurement view.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Fetch commands

## Syntax

`FETCh:LTE:CHSPepectrum:OBW?`

## Returns

<NR3>

## Examples

`FETCH:LTE:CHSPECTRUM:OBW?` might return `1.0851851250E+6`, indicating the Occupied Bandwidth is 1.09 MHz.

## FETCh:LTE:CHSPepectrum:POWer:CHANnel? (Query Only)

Returns the Channel Power value in LTE Channel Spectrum display. The result will be in dBm (by default) or in the units chosen from the Units tab of the Analysis Control panel.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Fetch commands

## Syntax

`FETCh:LTE:CHSPepectrum:POWer:CHANnel?`

## Returns

<NRf>

## Examples

`FETCH:LTE:CHSPECTRUM:POWER:CHANNEL?` might return `-14.9248560147`, indicating that the Channel Power is -14.92 dBm.

## FETCh:LTE:CHSPepectrum:SPECTrum? (Query Only)

Returns spectrum trace data of the LTE Channel Spectrum measurement.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Fetch commands

## Syntax

`FETCh:LTE:CHSPepectrum:SPECTrum?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of the data that follow.

<data(n)> is the amplitude in dBm at the  $n^{th}$  data point, 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

`FETCH:LTE:CHSPECTRUM:SPECTRUM?` might return `#43204 xxxx...` (3204-byte data) for the spectrum trace data of the LTE Channel Spectrum measurement.

## FETCh:LTE:CONStE:CELL:ID? (Query Only)

Queries the Cell ID value LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Fetch commands

### Syntax

`FETCh:LTE:CONStE:CELL:ID?`

### Returns

<NR1>

## Examples

`FETCH:LTE:CONStE:CELL:ID?` might return 112, indicating CELLID measurement number is 112.

## FETCh:LTE:CONStE:FREQuency:ERRor? (Query Only)

Queries Frequency Error value in Hz LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Fetch commands

### Syntax

`FETCh:LTE:CONStE:FREQuency:ERRor?`

### Returns

<NRf>

## Examples

`FETCH:LTE:CONStE:FREQUENcY:ERROR?` might return `-71.9780578613`, indicating frequency error value is `-71.9780578613` Hz.

## **FETCh:LTE:CONStE:GROUp:ID? (Query Only)**

Queries the Group ID value LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Fetch commands

### Syntax

`FETCh:LTE:CONStE:GROUp:ID?`

### Returns

`<NR1>`

## Examples

`FETCH:LTE:CONStE:GROUp:ID?` might return `160`, indicating Group ID measurement number is `160`.

## **FETCh:LTE:CONStE:SECTor:ID? (Query Only)**

Queries the Sector ID value for the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Fetch commands

### Syntax

`FETCh:LTE:CONStE:SECTor:ID?`

### Returns

`<NRf>`

## Examples

`FETCH:LTE:CONStE:SECTor:ID?` might return `2`, indicating the Sector ID value is `2`.



## **FETCh:LTE:PVTime:OFFSlot:POWer? (Query Only)**

Returns the TOff power measurement value in dBm/MHz for the LTE Power vs Time display.

### **Conditions**

Measurement view: LTE Power vs Time

### **Group**

Fetch commands

### **Syntax**

```
FETCh:LTE:PVTime:OFFSlot:POWer?
```

### **Returns**

<NR>

### **Examples**

FETCh:LTE:PVTIME:OFFSLOT:POWER? might return -76.11514587403, indicating the offslot power is -76.11514 dBm/MHz.

## **FETCh:LTE:PVTime:RESUlts:STATus**

Returns the pass or fail status for the LTE Power vs Time display.

### **Conditions**

Measurement view: LTE Power vs Time

### **Group**

Fetch commands

### **Syntax**

```
FETCh:LTE:PVTime:RESUlts:STATus
```

### **Returns**

0: means measurement status is FAIL.

1: means measurement status is PASS.

### **Examples**

FETCh:LTE:PVTIME:RESULTS:STATUS might return 1, indicating the measurement status is PASS.

## **FETCh:LTE:PVTime:TRACe:X? (Query Only)**

Returns the horizontal values (time in seconds) for the LTE Power vs. Time trace.

## Conditions

Measurement view: LTE Power vs Time

## Group

Fetch commands

## Syntax

FETCh:LTE:PVTime:TRACe:X?

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the  $n^{\text{th}}$  point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:LTE:PVTIME:TRACE:X? might return #43204 (3204-byte data), which represents the time in seconds (horizontal values).

## FETCh:LTE:PVTime:TRACe:XY? (Query Only)

Returns the horizontal value (time in seconds) and vertical value (power) for the LTE Power vs. Time trace.

## Conditions

Measurement view: LTE Power vs Time

## Group

Fetch commands

## Syntax

FETCh:LTE:PVTime:TRACe:XY?

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the  $n^{\text{th}}$  point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:LTE:PVTIME:TRACE:XY?` might return #43204 (3204-byte data), which represents the horizontal value (time in seconds) and vertical value (power) pair at the  $n^{\text{th}}$  data point.

## FETCh:LTE:PVTime:TRACe:Y? (Query Only)

Returns the vertical values (power) for the LTE Power vs. Time trace.

### Conditions

Measurement view: LTE Power vs Time

### Group

Fetch commands

### Syntax

`FETCh:LTE:PVTime:TRACe:Y?`

### Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the  $n^{\text{th}}$  point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:LTE:PVTIME:TRACE:Y?` might return #43204 (3204-byte data), which represents the vertical values (power).

## FETCh:MCPower:ADJacent:CHANnels? (Query Only)

Returns the power of adjacent channels in order of increasing frequency.

### Conditions

Measurement views: MCPR

### Group

Fetch commands

### Syntax

`FETCh:MCPower:ADJacent:CHANnels?`

### Arguments

None

## Returns

<acpr\_lower (n)>, ...<acpr\_lower (2)>, <acpr\_lower (1)>, <acpr\_upper (1)>, <acpr\_upper (2)>, ...<acpr\_upper (n)>

Where

<acpr\_lower (n)> is the ACPR for the lower channel #n in dB.

<acpr\_upper (n)> is the ACPR for the upper channel #n in dB.

To add a pair of upper and lower adjacent channels, use the [SENSe]:MCPower:CHANnel:ADJacent:ADD command.

## Examples

FETCH:MCPOWER:ADJACENT:CHANNELS? might return -4.420,-4.847,-4.316,-4.225, indicating (ACPR for the lower channel 2) = -4.420 dB, (ACPR for the lower channel 1) = -4.847 dB, (ACPR for the upper channel 1) = -4.316 dB, and (ACPR for the upper channel 2) = -4.225 dB.

## FETCh:MCPower:CHANnel:POWer? (Query Only)

Returns the reference power in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Fetch commands

### Syntax

FETCh:MCPower:CHANnel:POWer?

### Arguments

None

### Returns

<ref\_power>:<NRf> is the reference power in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

To select the power reference, use the [SENSe]:MCPower:RCHannels? commands.

## Examples

FETCH:MCPOWER:CHANNEL:POWER? might return 4.227, indicating that the reference power is 4.227 dBm.

## FETCh:MCPower:MAIN:CHANnels? (Query Only)

Returns the power of main channels in order of increasing frequency.

## Conditions

Measurement views: MCPR

## Group

Fetch commands

## Syntax

```
FETCh:MCPower:MAIN:CHANnels?
```

## Arguments

None

## Returns

```
<power_main(1)>,<power_main(2)>,...<power_main(n)>
```

Where

<power\_main(n)> is the power of main channel #n in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

To specify the main channels, use the [SENSe]:MCPower:CHANnel:MAIN commands.

## Examples

FETCH:MCPOWER:MAIN:CHANNELS? might return -2.420,-2.847,-2.316,-2.225, indicating (power of the main channel 1) = -2.420 dBm, (power of the main channel 2) = -2.847 dBm, (power of the main channel 3) = -2.316 dBm, and (power of the main channel 4) = -2.225 dBm.

## FETCh:MCPower:SPECTrum? (Query Only)

Returns spectrum trace data of the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Fetch commands

## Syntax

```
FETCh:MCPower:SPECTrum?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

## Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCH:MCPOWER:SPECTRUM? might return #43204xxxx... (3204-byte data) for the spectrum trace data of the MCPR measurement.

## FETCh:MERRor[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Fetch commands

## Syntax

FETCh:MERRor[:MEASview<y>]:FERRor?

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

FETCh:MERRor:MEASview2:FERRor? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

## FETCh:MERRor[:MEASview<y>]:PEAK? (Query only)

Returns the peak value in the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:MEERRor[:MEASview<y>]:PEAK?
```

## Related Commands

[FETCh:MEERRor\[:MEASview<y>\]:PINDEx?](#)

## Arguments

None

## Returns

<peak> : : <NRf> is the peak magnitude error value in percent (%).

## Examples

FETCh:MEERRor:MEASview2:PEAK? might return 1.57, indicating the peak magnitude error value is 1.57%.

## FETCh:MEERRor[:MEASview<y>]:PINDEx? (Query only)

Returns the time at the magnitude error peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:MEERRor[:MEASview<y>]:PINDEx?
```

## Related Commands

[FETCh:MEERRor\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

`<peak_time>::<NRf>` is the time at the magnitude error peak in symbol number.

The unit can be changed by the `[SENSE]:DDEMod:TIME:UNITs` command.

## Examples

`FETCh:MERRor:MEASview2:PINDEX?` might return 68.000, indicating that the Magnitude error peak is at symbol #68.

## FETCh:MERRor[:MEASview<y>]:RMS? (Query only)

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:MERRor[:MEASview<y>]:RMS?
```

## Arguments

None

## Returns

`<rms>::<NRf>` is the RMS magnitude error in percent (%).

## Examples

`FETCh:MERRor:MEASview2:RMS?` might return 0.382, indicating the RMS Magnitude error value is 0.382%.

## FETCh:MERRor[:MEASview<y>]:TRACe? (Query only)

Returns the Magnitude error versus Time trace data.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).



## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:MEERRor[:MEASview<y>]:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the magnitude error in percent (%) at the n<sup>th</sup> data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:MEERRor:MEASview2:TRACe? might return #42036xxxx... (2036-byte data) for the Magnitude error versus Time Measview 2.

## FETCh:NOISe:GAIN:TRACe<x>:RESult? (Query Only)

This query returns the gain value in mdB of the specified trace in the Gain display.

## Conditions

Measurement view: Gain

The parameter <x> = 1 to 3.

## Group

Fetch commands

## Syntax

```
FETCh:NOISe:GAIN:TRACe<x>:RESult?
```

## Returns

The gain value in mdB of the specified trace.

## Examples

`FETCH:NOISE:GAIN:TRACE1:RESULT?` might return `-88.7`, indicating that the gain is `-88.7` mdB.

## FETCh:NOISe:TABLE:VALue? (Query Only)

Returns Noise Table data. A total of 7 values per frequency (28 bytes per frequency) is returned. The number of frequencies returned depends on the settings.

### Conditions

Measurement view: Noise Table

### Group

Fetch commands

### Syntax

`FETCh:NOISe:TABLE:VALue?`

### Returns

Returns an array of floats (size of 4 bytes each) defined as:

- First frequency gain, noise figure, noise factor, y factor, noise temperature, power hot, power cold
- Second frequency gain, noise figure, noise factor, y factor, noise temperature, power hot, power cold

## Examples

`FETCH:NOISE:TABLE:VALUE?` might return `#3308`, indicating that the

## FETCh:NR:ACP:TAGGpower ? (Query Only)

Queries the Total Aggregate Power of the NR ACP display in the NR Summary View table.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:ACP:TAGGpower ?`

### Returns

`<NRf>`

## Examples

`FETCH:NR:ACP:TAGGPOWER ?` might return `-18.4524345398` dBm, indicating the Total Aggregate Power.

## **FETCh:NR:ACP:NO:OF:NR:OFFsets? (Query Only)**

Queries NR Offsets of the NR ACP display in the NR Summary View Table.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:NR:ACP:NO:OF:NR:OFFsets??
```

### **Returns**

<NRf>

### **Examples**

FETCh:NR:ACP:NO:OF:NR:OFFSETS? might return 6.0000000000, indicating the maximum offset is 6.

## **FETCh:NR:ACP:ADJChannel<x>:INTBw? (Query Only)**

Queries the bandwidth of the adjacent channel of the NR ACP in NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:NR:ACP:ADJChannel<x>:INTBw?
```

### **Returns**

<NRf>

### **Examples**

FETCh:NR:ACP:ADJChannel2:INTBw? might return 4.5150000000E+6 Hz, indicating the bandwidth of adjacent channel 2.

## **FETCh:NR:ACP:ADJChannel<x>:UABS? (Query Only)**

Queries the upper absolute value of the adjacent channel of the NR ACP in NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:ACP:ADJChannel<x>:UABS?`

## Returns

<NRf>

## Examples

`FETCh:NR:ACP:ADJChannel2:UABS?` might return -46.5615577698 dB, indicating the upper absolute value of adjacent channel 2.

## FETCh:NR:ACP:ADJChannel<x>:UREL ? (Query Only)

Queries the upper relative value of the adjacent channel of the NR ACP in NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:ACP:ADJChannel<x>:UREL?`

## Returns

<NRf>

## Examples

`FETCh:NR:ACP:ADJChannel1:UREL?` might return -47.5079994202 dB, indicating the upper relative value of adjacent channel 1.

## FETCh:NR:ACP:ADJChannel<x>:LABS? (Query Only)

Queries the lower absolute value of the adjacent channel of the NR ACP in NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:ACP:ADJChannel<x>:LABS?
```

## Returns

<NRf>

## Examples

FETCh:NR:ACP:ADJChannel1:LABS? might return -65.0139923096 dB, indicating the lower absolute value of adjacent channel 1.

## FETCh:NR:ACP:ADJChannel<x>:LREL ? (Query Only)

Queries the lower relative value of the adjacent channel of the NR ACP in NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:ACP:ADJChannel<x>:LREL?
```

## Returns

<NRf>

## Examples

FETCh:NR:ACP:ADJChannel2:LREL? might return -46.5615577698 dB, indicating the lower relative value of adjacent channel 2.

## FETCh:NR:ACP:ADJChannel<x>:FREQoffset ? (Query Only)

Queries the frequency offset of the adjacent channel of the NR ACP in the NR Summary View Table. <x> is the adjacent channel. Range is 1 to 6.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:ACP:ADJChannel<x>:FREQoffset?`

## Returns

<NRf>

## Examples

`FETCh:NR:ACP:ADJChannel2:FREQoffset?` might return `5.0000000000E+6` Hz, indicating the frequency offset of adjacent channel 2.

## **FETCh:NR:CHP:CARRier<x>:ABSPower? (Query Only)**

Queries the absolute power of each component carrier in the NR Channel Power in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:CHP:CARRier<x>:ABSPower?`

## Returns

<NRf>

## Examples

`FETCh:NR:CHP:CARRier6:ABSPower?` might return `-13.8919143677` dBm, indicating the absolute power of component carrier 6.

## **FETCh:NR:CHP:CARRier<x>:RELPower? (Query Only)**

Queries the relative power of each component carrier in the NR Channel Power in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:CHP:CARRier<x>:RELPower?`

---

## Returns

<NRf>

## Examples

FETCh:NR:CHP:CARRier3:RELPower? might return 0.0000, indicating the relative power of component carrier 3.

## FETCh:NR:CHP:TAGGpower? (Query Only)

Queries the total aggregate power of the NR Channel Power in the NR Summary View Table.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:CHP:TAGGpower?

## Returns

<NRf>

## Examples

FETCh:NR:CHP:TAGGPOWER? might return -13.8919143677 dBm, indicating the total aggregate power.

## FETCh:NR:EVM:ATSYmbol (Query Only)

Queries Symbol value for the selected trace in NR EVM.

## Conditions

Measurement view: NR EVM measurement view

## Group

Fetch commands

## Syntax

FETCh:NR:EVM:ATSYmbol

## Returns

<NRf>

## Examples

`FETCh:NR:EVM:ATSYmbol?` might return `9.9988253091E-3 %` indicating Symbol value for the selected trace.

## FETCh:NR:EVM:PEAK (Query Only)

Queries Peak value for the selected trace in NR EVM.

### Conditions

Measurement view: NR EVM measurement view

### Group

Fetch commands

### Syntax

`FETCh:NR:EVM:PEAK`

### Returns

<NRf>

## Examples

`FETCh:NR:EVM:PEAK?` might return `9.9988253091E-3 %` indicating Peak value for the selected trace.

## FETCh:NR:EVM:RMS (Query Only)

Queries RMS value for the selected trace in NR EVM.

### Conditions

Measurement view: NR EVM measurement view

### Group

Fetch commands

### Syntax

`FETCh:NR:EVM:RMS`

### Returns

<NRf>

## Examples

`FETCh:NR:EVM:RMS?` might return `9.9988253091E-3 %` indicating RMS value for the selected trace.



## **FETCh:NR:LINK:DIREction? (Query Only)**

Queries the link direction of the NR Constellation in NR Summary View Table.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:LINK:DIREction?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:LINK:DIRECTION? might return DOWN, indicating that the link direction is downlink.

## **FETCh:NR:MODAcc:CARRier<x>:CELL:ID? (Query Only)**

Queries the Cell ID of the NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:CELL:ID?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier5:CELL:ID? might return 1, indicating that the cell ID is 5.

## **FETCh:NR:MODAcc:CARRier<x>:OFFSet? (Query Only)**

Queries the carrier offset of the NR Constellation in NR Summary View Table.

<x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:MODAcc:CARRier<x>:OFFSet?`

## Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier2:OFFSet?` might return -184.740, indicating the offset value for component carrier 2 is -184.740 MHz.

## FETCh:NR:MODAcc:CARRier<x>:CREM? (Query Only)

Queries the Composite RMS EVM Mean Value of the NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:MODAcc:CARRier<x>:CREM?`

## Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier2:CREM?` might return 255.7628555223E-3, indicating that the component carrier 2 Composite RMS EVM Mean value is 0.256%.

## FETCh:NR:MODAcc:CARRier<x>:CPEM? (Query Only)

Queries the Composite Peak EVM Maximum Value of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:MODAcc:CARRier<x>:CPEM?
```

## Returns

<NRf>

## Examples

FETCh:NR:MODAcc:CARRier3:CPEM? might return 672.3846774548E-3, indicating the component carrier 3 Composite Peak EVM Maximum Value is 0.672%.

## FETCh:NR:MODAcc:CARRier<x>:CPESLI? (Query Only)

Queries the Composite Peak Slot Index of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:MODAcc:CARRier<x>:CPESLI?
```

## Returns

<NRf>

## Examples

FETCh:NR:MODAcc:CARRier1:CPESLI? might return 0, indicating the component carrier 1 Composite Peak EVM Slot Index is 0.

## FETCh:NR:MODAcc:CARRier<x>:CPESYI? (Query Only)

Queries the Composite Peak Symbol Index of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:MODAcc:CARRier<x>:CPESYI?`

## Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier2:CPESYI` might return 13, indicating component carrier 2 Composite Peak EVM Symbol Index is 13.

## FETCh:NR:MODAcc:CARRier<x>:CPESCI? (Query Only)

Queries the Composite Peak EVM Subcarrier Index of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:MODAcc:CARRier<x>:CPESCI?`

## Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier3:CPESCI?` might return 113, indicating the component carrier 3 Composite Peak EVM Subcarrier Index is 113.

## FETCh:NR:MODAcc:CARRier<x>:CCFME? (Query Only)

Queries the Composite Carrier Frequency Mean Error of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:MODAcc:CARRier<x>:CCFME?`

## Returns

<NRf>

## Examples

FETCh:NR:MODAcc:CARRier1:CCFME? might return 528.9274902344, indicating the composite carrier 1 Frequency Mean Error is 528.9274902344 Hz.

## FETCh:NR:MODAcc:CARRier<x>:CCIOOM? (Query Only)

Queries the Composite Carrier IQ Origin Offset Mean of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:MODAcc:CARRier<x>:CCIOOM?

## Returns

<NRf>

## Examples

FETCh:NR:MODAcc:CARRier2:CCIOOM? might return -70.8180387744, indicating composite carrier 2 IQ Origin Offset Mean is -70.8180387744 dBc.

## FETCh:NR:MODAcc:CARRier<x>:CCIGIM? (Query Only)

Queries the Composite Carrier IQ Gain Imbalance Mean of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:MODAcc:CARRier<x>:CCIGIM?

## Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier3:CCIqIM?` might return 0.001, indicating component carrier 3 Composite Carrier IQ Gain Imbalance Mean is 0.001 dB.

## **FETCh:NR:MODAcc:CARRier<x>:CCIqEM? (Query Only)**

Returns the Composite Carrier IQ Quadrature Error Mean of NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:MODAcc:CARRier<x>:CCIqEM?`

### Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier3:CCIqEM?` might return 19.3109996617E-3, indicating component carrier 3 Composite Carrier IQ Quadrature Error Mean is 19.3109996617E-3.

## **FETCh:NR:MODAcc:CARRier<x>:IBEM? (Query Only)**

Queries the In-Band Emission Range of the NR Constellation in NR Summary View Table. <x> is the component carrier. Range is 1 to 8.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:MODAcc:CARRier<x>:IBEM?`

### Returns

<NRf>

## Examples

`FETCh:NR:MODAcc:CARRier1:IBEM?` might return 9.9100E+37, indicating the component carrier 1 In-Band Emission Range.

## **FETCh:NR:MODAcc:CARRier<x>:DATA:PEAK (Query Only)**

Queries the SCH Data Peak EVM Maximum Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

```
FETCh:NR:MODAcc:CARRier<x>:DATA:PEAK?
```

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:DATA:PEAK? might return 672.3846774548E-3 ,indicating SCH Data Peak EVM Maximum Value is 0.672%.

## **FETCh:NR:MODAcc:CARRier<x>:DATA:RMS (Query Only)**

Queries the SCH Data RMS EVM Mean Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

```
FETCh:NR:MODAcc:CARRier<x>:DATA:RMS?
```

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:DATA:RMS? might return 286.3456029445E-3,indicating SCH Data RMS EVM Mean Value is 0.286%.

## **FETCh:NR:MODAcc:CARRier<x>:DMRS:PEAK (Query Only)**

Queries the SCH DMRS Peak EVM Maximum Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:DMRS:PEAK?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:DMRS:PEAK? might return 616.6561972350E-3, indicating SCH DMRS Peak EVM Maximum Value is 0.617%.

## **FETCh:NR:MODAcc:CARRier<x>:DMRS:RMS (Query Only)**

Queries the SCH DMRS EVM Mean Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:DMRS:RMS?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:DMRS:RMS? might return 251.3078739867E-3, indicating SCH DMRS EVM Value is 0.251%.



## **FETCh:NR:MODAcc:CARRier<x>:PREM (Query Only)**

Queries the Peak RMS EVM Maximum Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:PREM?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:PREM? might return 672.3846774548E-3 ,indicating Peak EVM Maximum Value is 0.672%.

## **FETCh:NR:MODAcc:CARRier<x>:PTRS: RMS (Query Only)**

Queries the SCH PTRS EVM Mean Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:PTRS:RMS?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:PTRS:RMS? might return 672.3846774548E-3 ,indicating SCH PTRS EVM MEAN Value is 0.672%.

## **FETCh:NR:MODAcc:CARRier<x>:PTRS:PEAK (Query Only)**

Queries the SCH PTRS Peak EVM Maximum Value of NR Constellation in NR Summary View Table display.

The parameter <x>=1 to 8, representing range of Component Carrier.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:MODAcc:CARRier<x>:PTRS:PEAK?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:MODAcc:CARRier1:PTRS:PEAK? might return 815.3068833053E-3 ,indicating SCH PTRS Peak EVM Maximum.

## **FETCh:NR:OBW:ABSPower (Query Only)**

Queries Absolute power of NR OBW display in NR Summary View Table display.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:OBW:ABSPower

### **Returns**

<NRf>

### **Examples**

FETCh:NR:OBW:ABSPower? might return -13.8897142410 dBm, indicating the Absolute power.

## **FETCh:NR:OBW:OCCUpiedbw (Query Only)**

Queries Occupied Bandwidth of NR OBW display in NR Summary View Table display.

## Conditions

Measurement view: NR Summary View Table measurement view

## Group

Fetch commands

## Syntax

```
FETCh:NR:OBW:OCCUpiedbw
```

## Returns

<NRf>

## Examples

FETCh:NR:OBW:OCCUpiedbw? might return 3.9506140488 MHz, indicating the Occupied Bandwidth.

## FETCh:NR:OBW:STARTfreq (Query Only)

Queries Start frequency of NR OBW display in NR Summary View Table display.

## Conditions

Measurement view: NR Summary View Table measurement view

## Group

Fetch commands

## Syntax

```
FETCh:NR:OBW:STARTfreq
```

## Returns

<NRf>

## Examples

FETCh:NR:OBW:STARTfreq? might return 998.0183831384E+6, indicating the Stop frequency.

## FETCh:NR:OBW:STOPfreq (Query Only)

Queries Stop frequency of NR OBW display in NR Summary View Table display.

## Conditions

Measurement view: NR Summary View Table measurement view

## Group

Fetch commands

### Syntax

FETCh:NR:OBW:STOPfreq

### Returns

<NRf>

### Examples

FETCh:NR:OBW:STOPfreq? might return 1.0019689972E+9 , indicating the Stop frequency.

## FETCh:NR:PVT:ABS:OFFpower:AFTR (Query Only)

Queries Absolute Off Power After NR Power vs Time display in NR Summary View Table display.

### Conditions

Measurement view: NR Summary View Table measurement view

### Group

Fetch commands

### Syntax

FETCh:NR:PVT:ABS:OFFpower:AFTR

### Returns

<NRf>

### Examples

FETCh:NR:PVT:ABS:OFFpower:AFTR? might return -62.4365692139 dBm, indicating the Absolute Off Power After.

## FETCh:NR:PVT:ABS:OFFpower:BFR (Query Only)

Queries Absolute Off Power Before of NR Power vs Time display in NR Summary View Table display.

### Conditions

Measurement view: NR Summary View Table measurement view

### Group

Fetch commands

### Syntax

FETCh:NR:PVT:ABS:OFFpower:BFR

### Returns

<NRf>

## Examples

`FETCh:NR:PVT:ABS:OFFpower:BFR?` might return -62.4902038574 dBm, indicating the Absolute Off Power Before.

## FETCh:NR:PVT:ABS:ONpower (Query Only)

Queries Absolute ON Power NR Power vs Time display in NR Summary View Table display.

### Conditions

Measurement view: NR Summary View Table measurement view

### Group

Fetch commands

### Syntax

`FETCh:NR:PVT:ABS:ONpower`

### Returns

<NRf>

## Examples

`FETCh:NR:PVT:ABS:ONpower?` might return -22.0409736633 dBm, indicating Absolute on power.

## FETCh:NR:PVT:BURStwidth (Query Only)

Queries Burst Width NR Power vs Time display in NR Summary View Table display.

### Conditions

Measurement view: NR Summary View Table measurement view

### Group

Fetch commands

### Syntax

`FETCh:NR:PVT:BURStwidth`

### Returns

<NRf>

## Examples

`FETCh:NR:PVT:BURStwidth?` might return 9.9988253091E-3 s, indicating Burst Width.

## **FETCh:NR:PVT:MEASurement:STATus (Query Only)**

Queries measurement status of NR Power vs Time display in NR Summary View Table display.

### **Conditions**

Measurement view: NR Summary View Table measurement view

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:PVT:MEASurement:STATus

### **Returns**

<NRf>

### **Examples**

FETCh:NR:PVT:MEASurement:STATus? might return "PASS", indicating the Power vs Time passed.

## **FETCh:NR:SEM:TAGGpower? (Query Only)**

Queries the Total Aggregate Power of the NR SEM in NR Summary View Table.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:SEM:TAGGpower?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:SEM:TAGGPOWER? might return -14.0847625732 dBm, indicating the Total Aggregate Power.

## **FETCh:NR:SEM:OFFSet<x>:STARTfreq? (Query Only)**

Queries the Start Frequency of the specified offset of the NR SEM display in NR Summary View Table. <x> is the offset. Range is 1 to 16.

### **Conditions**

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:SEM:OFFSet<x>:STARtfreq?
```

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet5:STARtfreq? might return -25.000000000E+3 Hz, indicating the Start Frequency of offset 5.

## FETCh:NR:SEM:OFFSet<x>:STOPfreq? (Query Only)

Queries the Stop Frequency of the specified offset of the NR SEM display in NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:SEM:OFFSet<x>:STOPfreq?
```

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet10:STOPlfreq? might return -975.000000000E+3 Hz, indicating the Stop Frequency of offset 10.

## FETCh:NR:SEM:OFFSet<x>:PEAK:ABS? (Query Only)

Queries the peak absolute value of specified offset of NR SEM display in NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:SEM:OFFSet<x>:PEAK:ABS?
```

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet1:PEAK:ABS? might return -65.5201115608 dBm, indicating the Peak Absolute value of offset 1.

## FETCh:NR:SEM:OFFSet<x>:PEAK:REL? (Query Only)

Queries the peak relative value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:SEM:OFFSet<x>:PEAK:REL?

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet1:PEAK:REL? might return -51.4353489876 dB, indicating the Peak Relative of offset 1.

## FETCh:NR:SEM:OFFSet<x>:PEAK:FREQ? (Query Only)

Queries the peak frequency of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:SEM:OFFSet<x>:PEAK:FREQ?

## Returns

<NRf>



## Examples

`FETCh:NR:SEM:OFFSet1:PEAK:FREQ?` might return 997.4753714650E+6 GHz, indicating the Peak Frequency of offset 1.

## FETCh:NR:SEM:OFFSet<x>:MARGin? (Query Only)

Queries the margin of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:SEM:OFFSet<x>:MARGin?`

### Returns

<NRf>

## Examples

`FETCh:NR:SEM:OFFSet3:MARGin?` might return -52.5201110840 dB, indicating the Margin of offset 3.

## FETCh:NR:SEM:OFFSet<x>:MARGin:ABS? (Query Only)

Queries the margin absolute value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:SEM:OFFSet<x>:MARGin:ABS?`

### Returns

<NRf>

## Examples

`FETCh:NR:SEM:OFFSet2:MARGin:ABS?` might return -51.4353489876 dBm, indicating the Margin Absolute value of offset 2.

## **FETCh:NR:SEM:OFFSet<x>:MARGin:REL? (Query Only)**

Queries the margin relative value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:SEM:OFFSet<x>:MARGin:REL?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:SEM:OFFSet1:MARGin:REL? might return -51.4353489876 dB, indicating the Margin Relative value of offset 1.

## **FETCh:NR:SEM:OFFSet<x>:MARGin:FREQ? (Query Only)**

Queries the margin frequency of the specified offset of the NR SEM display in the NR Summary View Table <x> is the offset. Range is 1 to 16.

### **Conditions**

Measurement view: NR Summary View Table

### **Group**

Fetch commands

### **Syntax**

FETCh:NR:SEM:OFFSet<x>:MARGin:FREQ?

### **Returns**

<NRf>

### **Examples**

FETCh:NR:SEM:OFFSet1:MARGin:FREQ? might return -51.4353489876 GHz, indicating the margin frequency of offset 1.

## **FETCh:NR:SEM:OFFSet<x>:INTEg:ABS? (Query Only)**

Queries the Integ Absolute value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:SEM:OFFSet<x>:INTEg:ABS?
```

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet3:INTEg:ABS? might return -51.4353489876 dBm, indicating the Integ Absolute value of offset 3.

## FETCh:NR:SEM:OFFSet<x>:INTEg:REL? (Query Only)

Queries the Integ Relative value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

```
FETCh:NR:SEM:OFFSet<x>:INTEg:REL?
```

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet2:INTEg:REL? might return -51.4353489876 dB, indicating the Integ Relative value of offset 2.

## FETCh:NR:SEM:OFFSet<x>:RBW? (Query Only)

Queries the RBW of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:SEM:OFFSet<x>:RBW?

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet1:RBW? might return -51.4353489876 kHz, indicating the RBW of offset 1.

## FETCh:NR:SEM:OFFSet<x>:ABSLimit:START? (Query Only)

Queries the Absolute Start value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:SEM:OFFSet<x>:ABSLimit:START?

## Returns

<NRf>

## Examples

FETCh:NR:SEM:OFFSet2:ABSLimit:START? might return -51.4353489876 dBm, indicating the Absolute Start of offset 2.

## FETCh:NR:SEM:OFFSet<x>:ABSLimit:STOP? (Query Only)

Queries the Absolute Stop value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

FETCh:NR:SEM:OFFSet<x>:ABSLimit:STOP?

## Returns

<NRf>

## Examples

`FETCh:NR:SEM:OFFSet2:ABSLimit:START?` might return -51.4353489876 dBm, indicating the Absolute Stop value of offset 2.

## FETCh:NR:SEM:OFFSet<x>:RELLimit:START? (Query Only)

Queries the Relative Start value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:SEM:OFFSet<x>:RELLimit:START?`

## Returns

<NRf>

## Examples

`FETCh:NR:SEM:OFFSet2:RELLimit:START?` might return -51.4353489876 dBm, indicating the Relative Start value of offset 2.

## FETCh:NR:SEM:OFFSet<x>:RELLimit:STOP? (Query Only)

Queries the Relative Stop value of the specified offset of the NR SEM display in the NR Summary View Table. <x> is the offset. Range is 1 to 16.

## Conditions

Measurement view: NR Summary View Table

## Group

Fetch commands

## Syntax

`FETCh:NR:SEM:OFFSet<x>:RELLimit:STOP?`

## Returns

<NRf>

## Examples

`FETCh:NR:SEM:OFFSet2:RELLimit:STOP?` might return `-51.4353489876` dBm, indicating the Relative Stop value of offset 2.

## FETCh:NR:SUBblock<x>:NUMCcs? (Query Only)

Queries the number of component carriers in the specified subblock of the NR Constellation in the NR Summary View Table. <x> is the subblock. Range is 1.

### Conditions

Measurement view: NR Summary View Table

### Group

Fetch commands

### Syntax

`FETCh:NR:SUBblock<x>:NUMCcs?`

### Returns

<NRf>

## Examples

`FETCh:NR:SUBblock1:NUMCcs?` might return 8, indicating that the number of component carriers in subblock 1 is 8.

## FETCh:OBWidth:BOBW:FREQuency:ERRor (Query Only)

Returns the frequency error in the Bluetooth 20dB BW display.

### Conditions

Measurement view: Bluetooth 20dB BW

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:BOBW:FREQuency:ERRor?`

### Arguments

None

### Returns

<NRf>

## Examples

`FETCH:OBWIDTH:BOBW:FREQUENCY:ERROR` might return -999.1 kHz, indicating the frequency error is -999.1 kHz.

## FETCh:OBWidth:BOBW:RESUlts:STATus (Query Only)

Returns the test result as Pass or Fail in the Bluetooth 20dB BW display.

### Conditions

Measurement views: Bluetooth 20dB BW

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:BOBW:RESUlts:STATus`

### Arguments

None

### Returns

PASS means the test passed.

FAIL means the test failed.

INVALID means there was an error and the test did not complete.

## Examples

`FETCH:OBWIDTH:BOBW:RESULTS:STATUS` might return Pass or Fail or Invalid for Bluetooth 20dB BW display.

## FETCh:OBWidth:BOBW:XDBB:POWer (Query Only)

Returns the x dB BW Ref Power value for the selected bandwidth as shown in the Bluetooth 20dB BW display.

### Conditions

Measurement view: Bluetooth 20dB BW

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:BOBW:XDBB:POWer?`

### Returns

<NR3>

## Examples

`FETCH:OBWIDTH:BOBW:XDBB:POWER` might return `-9.0197525024`, indicating that the x dB reference power for the selected bandwidth is `-9.0197525024` dB.

## FETCh:OBWidth:BOBW:XDBBandwidth? (Query Only)

Returns the difference between the higher and lower frequency points corresponding to the value where it is X dB less from the value at the center frequency.

Set the measurement direction using the appropriate inward or outward command before issuing this command.

## Conditions

Measurement views: Bluetooth 20 dB Bandwith

## Group

Fetch commands

## Syntax

`FETCh:OBWidth:BOBW:XDBBandwidth?`

## Related Commands

`FETCh:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?`,  
`FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth?`

## Arguments

None

## Returns

`<value> ::= <NRf>` the X dBm bandwidth in Hz.

## Examples

`FETCH:OBWIDTH:BOBW:XDBBANDWIDTH?` might return `89.12000E+3` indicating the difference between the higher and lower frequency points is 89.12 kHz.

## FETCh:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth? (Query Only)

Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured inwards (coming from the ends to the center frequency).

## Conditions

Measurement views: Bluetooth 20 dB Bandwith

## Group

Fetch commands



## Syntax

```
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?
```

## Related Commands

```
FETCh:OBWidth:BOBW:XDBBANDwidth?, FETCh:OBWidth:BOBW: XDBMbandwidth:OUT:BANDwidth?
```

## Arguments

None

## Returns

<value> ::= <NRf> the input X dBm value in Hz.

## Examples

FETCh:OBWidth:BOBW:XDBMBANDwidth:IN:BANDwidth? might return 1.251840000E+6 indicating that the bandwidth corresponding to the input XdBm value is 1.2518 MHz measured in the inward direction.

## FETCh:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQuency? (Query Only)

Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the inward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Fetch commands

## Syntax

```
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQuency?
```

## Related Commands

```
FETCh:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQuency?
```

## Arguments

None

## Returns

<value> ::= <NRf> the left frequency in Hz.

## Examples

FETCh:OBWidth:BOBW:XDBMBANDwidth:IN:LEFT:FREQUENCY? might return 2.401E+9 indicating that the frequency corresponding to the X dbm value to the left of the center frequency is 2.401GHz.

## **FETCh:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQuency? (Query Only)**

Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the inward direction.

### **Conditions**

Measurement views: Bluetooth 20 dB Bandwidth

### **Group**

Fetch commands

### **Syntax**

FETCh:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQuency?

### **Related Commands**

FETCh:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQuency?

### **Arguments**

None

### **Returns**

<value> ::= <NRf> the right frequency in Hz.

### **Examples**

FETCh:OBWidth:BOBW:XDBMBANDWIDTH:IN:RIGHT:FREQUENCY? might return 2.445E+9 indicating that the frequency corresponding to the XdBm value to the right of the center the frequency is 2.445 GHz.

## **FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth? (Query Only)**

Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured outwards (from the center frequency to the ends).

### **Conditions**

Measurement views: Bluetooth 20 dB Bandwidth

### **Group**

Fetch commands

### **Syntax**

FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth?

### **Related Commands**

FETCh:OBWidth:BOBW:XDBBANDwidth?, FETCh:OBWidth:BOBW: XDBMbandwidth:IN:BANDwidth?

## Arguments

None

## Returns

<value> ::= <NRf> the input X dBm value in Hz.

## Examples

`FETCH:OBWIDTH:BOBW:XDBMBANDWIDTH:OUT:BANDWIDTH?` might return `1.251840000E+6` indicating that the bandwidth corresponding to the output XdBm value is 1.2518 MHz measured in the outward direction.

## **FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency? (Query Only)**

Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the outward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Fetch commands

## Syntax

`FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency?`

## Related Commands

`FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency?`

## Arguments

None

## Returns

<value> ::= <NRf> the left frequency in Hz.

## Examples

`FETCH:OBWIDTH:BOBW:XDBMBANDWIDTH:OUT:LEFT:FREQUENCY?` might return `2.401E+9` indicating that the frequency corresponding to the X dbm value to the left of the center frequency is 2.401GHz.

## **FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency? (Query Only)**

Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the outward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency?

## Related Commands

FETCh:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency?

## Arguments

None

## Returns

<value> ::= <NRf> the Right frequency in Hz.

## Examples

FETCh:OBWidth:BOBW:XDBMBANDWIDTH:OUT:RIGHT:FREQUENCY? might return 2.445E+9 indicating that the frequency corresponding to the XdBm value to the right of the center the frequency is 2.445 GHz.

## FETCh:OBWidth:FREQuency:ERRor? (Query Only)

Returns the frequency error in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:FREQuency:ERRor?

## Arguments

None

## Returns

<freq\_error>:::<NRf> is the frequency error in Hz.

## Examples

FETCh:OBWidth:FREQUENCY:ERROR? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

## FETCh:OBWidth:OBWidth:BANDwidth? (Query Only)

Returns the occupied bandwidth in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

```
FETCh:OBWidth:OBWidth:BANDwidth?
```

### Arguments

None

### Returns

<OBW>::<NRf> is the occupied bandwidth in Hz.

### Examples

FETCh:OBWidth:OBWidth:BANDwidth? might return 4.0E+6, indicating the occupied bandwidth is 4 MHz.

## FETCh:OBWidth:OBWidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the occupied bandwidth.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

```
FETCh:OBWidth:OBWidth:LEFT:FREQuency?
```

### Related Commands

```
FETCh:OBWidth:OBWidth:RIGHT:FREQuency?
```

### Arguments

None

### Returns

<OBW\_left\_freq>::<NRf> is the left frequency in Hz.

## Examples

`FETCH:OBWIDTH:OBWIDTH:LEFT:FREQUENCY?` might return 1.498E+9, indicating the left frequency is 1.498 GHz.

## **FETCh:OBWidth:OBWidth:LEFT:LEVel? (Query Only)**

Returns the level at the left frequency of the occupied bandwidth.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:OBWidth:LEFT:LEVel?`

### Related Commands

`FETCh:OBWidth:OBWidth:RIGHT:LEVel?`

### Arguments

None

### Returns

`<OBW_left_level>::<NRf>` is the level at the left frequency in dB.

## Examples

`FETCH:OBWIDTH:OBWIDTH:LEFT:LEVEL?` might return -23.5, indicating the level at the left frequency is -23.5 dB.

## **FETCh:OBWidth:OBWidth:POWer? (Query Only)**

Returns the reference power in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:OBWidth:POWer?`

### Arguments

None

## Returns

`<OBW_ref_power>::<NRf>` is the reference power in dBm. The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCH:OBWIDTH:OBWIDTH:POWER?` might return `-10.0`, indicating the reference power is `-10` dBm.

## FETCh:OBWidth:OBWidth:RIGHT:FREQuency? (Query Only)

Returns the right (higher) frequency of the occupied bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

`FETCh:OBWidth:OBWidth:RIGHT:FREQuency?`

## Related Commands

`FETCh:OBWidth:OBWidth:LEFT:FREQuency?`

## Arguments

None

## Returns

`<OBW_right_freq>::<NRf>` is the right frequency in Hz.

## Examples

`FETCH:OBWIDTH:OBWIDTH:RIGHT:FREQUENCY?` might return `1.502E+9`, indicating the right frequency is `1.502` GHz.

## FETCh:OBWidth:OBWidth:RIGHT:LEVel? (Query Only)

Returns the level at the right frequency of the occupied bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:OBWidth:RIGHT:LEVel?

## Related Commands

FETCh:OBWidth:OBWidth:LEFT:LEVel?

## Arguments

None

## Returns

<OBW\_right\_level>::<Nrf> is the level at the right frequency in dB.

## Examples

FETCH:OBWIDTH:OBWIDTH:RIGHT:LEVEL? might return -23.5, indicating the level at the right frequency is -23.5 dB.

## FETCh:OBWidth:SPECTrum? (Query Only)

Returns spectrum trace data of the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:SPECTrum?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSE]:POWER:UNITS command.



## Examples

`FETCH:OBWIDTH:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

## **FETCh:OBWidth:XDBBandwidth:BANDwidth? (Query Only)**

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:XDBBandwidth:BANDwidth?`

### Arguments

None

### Returns

`<xdB BW> : : <NRf>` is the x dB bandwidth in Hz.

## Examples

`FETCH:OBWIDTH:XDBBANDWIDTH:BANDWIDTH?` might return `2.0E+6`, indicating the x dB bandwidth is 2 MHz.

## **FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency? (Query Only)**

Returns the left (lower) frequency of the x dB bandwidth.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Fetch commands

### Syntax

`FETCh:OBWidth:XDBBandwidth:LEFT:FREQuency?`

### Related Commands

`FETCh:OBWidth:XDBBandwidth:RIGHT:FREQuency?`

## Arguments

None

## Returns

<xdBW\_left\_freq>::<Nrf> is the left frequency in Hz.

## Examples

FETCH:OBWIDTH:XDBBANDWIDTH:LEFT:FREQUENCY? might return 1.498E+9, indicating the left frequency is 1.498 GHz.

## FETCh:OBWidth:XDBBandwidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the x dB bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?

## Related Commands

FETCh:OBWidth:XDBBandwidth:RIGHT:LEVel?

## Arguments

None

## Returns

<xdBW\_left\_level>::<Nrf> is the level at the left frequency in dB.

## Examples

FETCH:OBWIDTH:XDBBANDWIDTH:LEFT:LEVEL? might return -23.5, indicating the level at the left frequency is -23.5 dB.

## FETCh:OBWidth:XDBBandwidth:POWer? (Query Only)

Returns the reference power in the x dB bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

```
FETCh:OBWidth:XDBBandwidth:POWer?
```

## Arguments

None

## Returns

<xdbBW\_ref\_power>::<NRf> is the reference power in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:OBWIDTH:XDBBANDWIDTH:POWER? might return -10.0, indicating the reference power is -10 dBm.

## FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY? (Query Only)

Returns the right (higher) frequency of the x dB bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

```
FETCh:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?
```

## Related Commands

```
FETCh:OBWidth:XDBBandwidth:LEFT:FREQUENCY?
```

## Arguments

None

## Returns

<xdbBW\_right\_freq>::<NRf> is the right frequency in Hz.

## Examples

FETCh:OBWIDTH:XDBBANDWIDTH:RIGHT:FREQUENCY? might return 1.502E+9, indicating the right frequency is 1.502 GHz.

## FETCh:OBWidth:XDBBandwidth:RIGHT:LEVEL? (Query Only)

Returns the level at the right frequency of the x dB bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Fetch commands

## Syntax

FETCh:OBWidth:XDBBandwidth:RIGHT:LEVel?

## Related Commands

FETCh:OBWidth:XDBBandwidth:LEFT:LEVel?

## Arguments

None

## Returns

<xdBW\_right\_level>::<NRf> is the level at the right frequency in dB.

## Examples

FETCh:OBWidth:XDBBANDWIDTH:RIGHT:LEVEL? might return -23.5, indicating the level at the right frequency is -23.5 dB.

## FETCh:OFDM:APOWer? (Query Only)

Returns the average power in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

FETCh:OFDM:APOWer?

## Arguments

None

## Returns

<apower>::<NRf> the average power in dB.

## Examples

FETCh:OFDM:APOWER? might return -23.4584459235 indicating the average power is -23.46 dB.

## **FETCh:OFDM:APOWer:PEAK? (Query Only)**

Returns the peak-to-average power in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:APOWer:PEAK?
```

### **Arguments**

None

### **Returns**

<power\_peak>::<NRf> is the peak-to-average power in dBm.

### **Examples**

FETCh:OFDM:APOWer:PEAK? might return 10.4140096289 indicating that peak-to-average power is 10.41 dBm.

## **FETCh:OFDM:CONSte:MAGNitude? (Query Only)**

Returns the constellation magnitude data for the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:CONSte:MAGNitude?
```

### **Related Commands**

```
FETCh:OFDM:CONSte:PHASe?
```

```
FETCh:OFDM:CONSte:TYPE?
```

```
FETCh:OFDM:CONSte:VALue?
```

### **Arguments**

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:OFDM:CONStE:MAGNITUDE? might return #510400xxxx... (10400-byte data) for the measurement.

## FETCh:OFDM:CONStE:PHASe? (Query Only)

Returns the constellation phase values of the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

FETCh:OFDM:CONStE:PHASe?

## Related Commands

FETCh:OFDM:CONStE:MAGNitude?

FETCh:OFDM:CONStE:TYPE?

FETCh:OFDM:CONStE:VALue?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:OFDM:CONStE:PHASe? might return #510400xxxx... (10400-byte data) for the measurement.

## FETCh:OFDM:CONStE:TYPE? (Query Only)

Returns the constellation context value of the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Fetch commands

### Syntax

```
FETCh:OFDM:CONStE:TYPE?
```

### Related Commands

```
FETCh:OFDM:CONStE:MAGNitude?
```

```
FETCh:OFDM:CONStE:PHASe?
```

```
FETCh:OFDM:CONStE:VALue?
```

### Arguments

None

### Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers. The integers must be translated to the proper context as listed below.

Integer value	Type (context)
0	Pilot
1	Data
2	Unused or Null

### Examples

FETCh:OFDM:CONStE:TYPE? might return #41352 (1352-byte data) for the measurement. The actual data must be decoded to the context type using the table above.

## FETCh:OFDM:CONStE:VALue? (Query Only)

Returns the constellation value (decoded symbol) of the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:CONStE:VALue?`

## Related Commands

`FETCh:OFDM:CONStE:MAGNitude?`

`FETCh:OFDM:CONStE:PHASe?`

`FETCh:OFDM:CONStE:TYPE?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the nth data point in 16-bit little endian integers.

## Examples

`FETCh:OFDM:CONStE:VALue?` might return `#41352` indicating 1352 bytes of data for the measurement.

## FETCh:OFDM:CPE? (Query Only)

Returns the Common Pilot Error magnitude for the OFDM measurement

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:CPE?`

## Arguments

None



## Returns

<NRf> the RMS magnitude error in percent.

## Examples

`FETCH:OFDM:CPE?` might return 3.7868041505 indicating that the CPE is 3.787%.

## FETCh:OFDM:CRESpOse:MAGNitude? (Query Only)

Returns the channel response magnitude data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:CRESpOse:MAGNitude?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:OFDM:CRESPONSE:MAGNITUDE?` might return `#3804xxxx... (804 byte data)` for the measurement.

## FETCh:OFDM:CRESpOse:PHASe? (Query Only)

Returns the channel response phase data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:CRESpOse:PHASe?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the *n*th data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:OFDM:CRESPONSE:PHASE?` might return `#3804xxxx... (804 byte data)` for the measurement.

## FETCh:OFDM:EVM:PEAK:DECibel:ALL? (Query Only)

Returns the peak EVM data for all subcarriers in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:EVM:PEAK:DECibel:ALL?`

## Arguments

None

## Returns

`<NRf>` data in dB.

## Examples

`FETCh:OFDM:EVM:PEAK:DECIBEL:ALL?` might return `-20.1872549032` indicating the data is `-20.19` dB for the measurement.

## FETCh:OFDM:EVM:PEAK:DECibel:DATA? (Query Only)

Returns the peak EVM data for the data subcarriers in the OFDM measurement.

---

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:EVM:PEAK:DECibel:DATA?
```

## Arguments

None

## Returns

<NRf> data in dB.

## Examples

FETCh:OFDM:EVM:PEAK:DECIBEL:DATA? might return -20.1872549032 indicating the data is -20.19 dB for the measurement.

## FETCh:OFDM:EVM:PEAK:DECibel:PILOts? (Query Only)

Returns the peak EVM data for the pilot subcarriers in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:EVM:PEAK:DECibel:PILOts?
```

## Arguments

None

## Returns

<NRf> data in dB.

## Examples

FETCh:OFDM:EVM:PEAK:DECIBEL:PILOTS? might return -20.2872549032 indicating the data is -20.29 dB for the measurement.

## **FETCh:OFDM:EVM:PEAK:PERCent:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:PERCent:ALL?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:PEAK:PERCENT:ALL? might return 19.3223863840 indicating the data is 19.322% for the measurement.

## **FETCh:OFDM:EVM:PEAK:PERCent:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:PERCent:DATA?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:PEAK:PERCENT:DATA? might return 19.3223863840 indicating the data is 19.322% for the measurement.

## **FETCh:OFDM:EVM:PEAK:PERCent:PILOts? (Query Only)**

Returns the peak EVM data for the pilot subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:EVM:PEAK:PERCent:PILOts?
```

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:PEAK:PERCENT:PILOTS? might return 10.8543863840 indicating the data is 10.854% for the measurement.

## **FETCh:OFDM:EVM:PEAK:SCARrier:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:EVM:PEAK:SCARrier:ALL?
```

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SCARRIER:ALL? might return 99.00000 indicating the data is 99 for the measurement.

## **FETCh:OFDM:EVM:PEAK:SCARrier:DATA? (Query Only)**

Returns the peak EVM data for all data subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:SCARrier:DATA?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SCARrier:DATA? might return 99.00000 indicating the data is 99 for the measurement.

## **FETCh:OFDM:EVM:PEAK:SCARrier:PILOts? (Query Only)**

Returns the peak EVM data for the pilot subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:SCARrier:PILOts?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SCARrier:PILOts? might return 88.00000 indicating the data is 88 for the measurement

---

## **FETCh:OFDM:EVM:PEAK:SYMBOL:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:SYMBOL:ALL?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SYMBOL:ALL? might return 2.00000 indicating the data is 2 for the measurement.

## **FETCh:OFDM:EVM:PEAK:SYMBOL:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:SYMBOL:DATA?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SYMBOL:DATA? might return 2.00000 indicating the data is 2 for the measurement.

## **FETCh:OFDM:EVM:PEAK:SYMBol:PILots? (Query Only)**

Returns the peak EVM data for the pilot subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:PEAK:SYMBol:PILots?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

FETCh:OFDM:EVM:PEAK:SYMBOL:PILOTS? might return 1.00000 indicating the data is 1 for the measurement.

## **FETCh:OFDM:EVM:RMS:DECibel:ALL? (Query Only)**

Returns the RMS EVM data for all subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:RMS:DECibel:ALL?

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

FETCh:OFDM:EVM:RMS:DECIBEL:ALL? might return -26.9012093267 indicating the data is -26.90 for the measurement.



## **FETCh:OFDM:EVM:RMS:DECibel:DATA? (Query Only)**

Returns the RMS EVM data for the data subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:EVM:RMS:DECibel:DATA?
```

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

FETCh:OFDM:EVM:RMS:DECIBEL:DATA? might return -26.8477116269 indicating the data is -26.85 for the measurement.

## **FETCh:OFDM:EVM:RMS:DECibel:PILOts? (Query Only)**

Returns the RMS EVM data for the pilot subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:EVM:RMS:DECibel:PILOts?
```

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

FETCh:OFDM:EVM:RMS:DECIBEL:PILOTS? might return -33.0589143032 indicating the data is -33.06 dB.

## **FETCh:OFDM:EVM:RMS:PERCent:ALL? (Query Only)**

Returns the peak RMS data for all subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:RMS:PERCent:ALL?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:RMS:PERCENT:ALL? might return 225.0743627548 indicating the data is 225.074% for the measurement.

## **FETCh:OFDM:EVM:RMS:PERCent:DATA? (Query Only)**

Returns the peak RMS data for the data subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:RMS:PERCent:DATA?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:RMS:PERCENT:DATA? might return 260.6973409653 indicating that the data is 260.697% for the measurement.

## **FETCh:OFDM:EVM:RMS:PERCent:PILOts? (Query Only)**

Returns the peak RMS data for the pilot subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:RMS:PERCent:PILOts?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

FETCh:OFDM:EVM:RMS:PERCENT:PILOTS? might return 210.8103863840 indicating the data is 210.810% for the measurement.

## **FETCh:OFDM:EVM:TRACe<x>? (Query Only)**

Returns the EVM trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

FETCh:OFDM:EVM:TRACe<x>?

### **Arguments**

None

### **Returns**

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:OFDM:EVM:TRACE2?` might return `#252xxxx. . .` (52 byte data) for the Average versus Symbols measurement.

## FETCH:OFDM:FERRor? (Query Only)

Returns the Frequency error reading for the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Fetch commands

### Syntax

`FETCH:OFDM:FERRor?`

### Arguments

None

### Returns

<NRf> frequency error.

## Examples

`FETCH:OFDM:FERROR?` might return `82.8617142098E+3` indicating the frequency error was 82.86 kHz.

## FETCH:OFDM:FLATness:PASS? (Query Only)

Queries whether the average power levels of subcarriers across the signal bandwidth on the OFDM Spectral Flatness display remain within the limits defined for a particular standard.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Fetch commands

## Syntax

```
FETCh:OFDM:FLATness:PASS?
```

## Returns

Returns "1" in the case that all data remains within the defined limits, and "0" otherwise.

## Examples

FETCh:OFDM:FLAT:PASS? might return 1, which means that the average power levels of subcarriers across the signal bandwidth remain within the defined limits.

## FETCh:OFDM:FLATness:RESult? (Query Only)

Returns the summary results of each segment of the OFDM Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:FLATness:RESult?
```

## Returns

The data is returned as 12 comma delineated text values. The specific values are :

- minimum subcarrier within segment 1
- minimum subcarrier average energy deviation from the global average within segment 1
- deviation the of minimum subcarrier average from the allowed value in segment 1
- maximum subcarrier within segment 1
- maximum subcarrier average energy deviation from the global average within segment 1
- deviation of the maximum subcarrier average from the allowed value in segment 1
- minimum subcarrier within segment 2
- minimum subcarrier average energy deviation from the global average within segment 2
- deviation the of minimum subcarrier average from the allowed value in segment 2
- maximum subcarrier within segment 2
- maximum subcarrier average energy deviation from the global average within segment 2
- deviation of the maximum subcarrier average from the allowed value in segment 2

## Examples

FETCh:OFDM:FLAT:RES? might return -6.0000000000,-17.0626174659E-3,3.9829373825,12.0000000000,19.979260861 which represents the summary results of each segment of the OFDM Spectral Flatness display.

## **FETCh:OFDM:FLATness:TRACe<x>? (Query Only)**

Returns the OFDM Spectral Flatness trace data. When <x> is 1, the parameter is Matrix (symbol deviation per subcarrier). When <x> is 2, the parameter is Average Deviation vs Subcarrier.

### **Conditions**

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:FLATness:TRACe<x>?
```

### **Returns**

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

### **Examples**

FETCh:OFDM:FLAT:TRACE2? might return #510400.. (10400-byte data) which represents the data for the Average Deviation vs. Subcarrier trace.

## **FETCh:OFDM:GAIN:IMBalance? (Query Only)**

Returns the gain imbalance for the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:OFDM:GAIN:IMBalance?
```

### **Arguments**

None

## Returns

<NRf> gain imbalance in dB.

## Examples

`FETCH:OFDM:GAIN:IMBALANCE?` might return `-57.746E-3` indicating that the gain imbalance is `-0.057746` dB.

## **FETCh:OFDM:IQ:ORIGin:OFFSet? (Query Only)**

Returns the IQ origin offset for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:IQ:ORIGin:OFFSet?`

## Arguments

None

## Returns

<NRf> gain imbalance in dB.

## Examples

`FETCH:OFDM:IQ:ORIGIN:OFFSET?` might return `-53.47017` indicating that the IQ origin offset is `-53.47` dB.

## **FETCh:OFDM:MERRor:TRACe<x>? (Query Only)**

Returns the magnitude error trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:MERRor:TRACe<x>?`

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:OFDM:MERROR:TRACE2` might return `#3108xxxx...` (108-byte data) for the average versus symbols trace measurement.

## FETCh:OFDM:PACKet:DIRection? (Query Only)

Returns the direction of the packet in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:PACKet:DIRection?`

## Arguments

None

## Returns

One of the following values.

`UP` specifies that the packet is in the up direction.

`DOWN` specifies that the packet is in the down direction.

`BOTH` specifies that the packet is both directions.

## Examples

`FETCH:OFDM:PACKET:DIRECTION?` might return `DOWN` indicating that the packet was in the down direction.

## FETCh:OFDM:PERRor:TRACe<x>? (Query Only)

Returns the phase error trace data for the OFDM measurement.



When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:PERROr:TRACe<x>?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:OFDM:PERROr:TRACe2 might return #3100xxxx. . . (100-byte data) for the average versus symbols trace measurement.

## FETCh:OFDM:POWer:TRACe<x>? (Query Only)

Returns the power trace data for the OFDM measurement. When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols. When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:POWer:TRACe<x>?
```

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:OFDM:POWER:TRACE2 might return #296xxxx. . . (96-byte data) for the average versus symbols trace measurement.

## FETCh:OFDM:QUADrature:OFFSet? (Query Only)

Returns the quadrature offset in the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Fetch commands

### Syntax

FETCh:OFDM:QUADrature:OFFSet?

### Arguments

None

### Returns

<NRf> quadrature offset in degrees.

### Examples

FETCH:OFDM:QUADRATURE:OFFSET? might return -99.9 indicating that the offset is -99°.

## FETCh:OFDM:SCARriers? (Query Only)

Returns the number of subcarriers in the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Fetch commands

---

## Syntax

`FETCh:OFDM:SCARriers?`

## Arguments

None

## Returns

<NR1> number of subcarriers.

## Examples

`FETCH:OFDM:SCARRIERS?` might return 200.0000 indicating that there are 200 subcarriers in the measurement.

## FETCh:OFDM:SCARriers:SPACing? (Query Only)

Returns the subcarrier spacing for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

`FETCh:OFDM:SCARriers:SPACing?`

## Arguments

None

## Returns

<NRf> subcarrier spacing.

## Examples

`FETCH:OFDM:SCARRIERS:SPACING?` might return 90.000E+3 indicating that the subcarrier spacing is 90.000 kHz.

## FETCh:OFDM:STABLE:VALUe? (Query Only)

Returns the number of bytes in the symbol table for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

FEtCh:OFDM:STABle:VALUe?

## Arguments

None

## Returns

#<num\_digit><num\_byte>

Where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data.

## Examples

FEtCh:OFDM:STABle:VALUe? might return #43848 indicating that there are 3848 bytes in the symbol table.

## FETCh:OFDM:SYMBol:CERRor? (Query Only)

Returns the symbol clock error for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

FEtCh:OFDM:SYMBol:CERRor?

## Arguments

None

## Returns

<NRf> symbol clock error in ppm.

## Examples

FEtCh:OFDM:SYMBol:CERRor? might return 422.7135479929 indicating that the symbol clock error was 422.714 ppm.

## FETCh:OFDM:SYMBol:COUNT? (Query Only)

Returns the number of symbols for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Fetch commands

## Syntax

```
FETCh:OFDM:SYMBOL:COUNT?
```

## Arguments

None

## Returns

<NR1> number of symbols in the measurement.

## Examples

FETCH:OFDM:SYMBOL:COUNT? might return 25.00000 indicating that there were 25 symbols for the measurement.

## FETCh:P25:CONStE:FERRor? (Query Only)

Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency of the analyzer.

When Frequency Error is in Auto mode, this query returns the frequency error value. When in Manual mode, it returns the frequency offset.

## Conditions

Measurement view: P25 Constellation

## Group

Fetch commands

## Syntax

```
FETCh:P25:CONStE:FERRor?
```

## Returns

<freq\_error>::<NRf>, which is the frequency error in Hz.

## Examples

FETCH:P25:CONStE:FERROR? might return -10.7E+3, which is a frequency error of -10.7 kHz.

## FETCh:P25:CONStE:TRACe? (Query Only)

Returns the P25 Constellation display trace data.

## Conditions

Measurement view: P25 Constellation

## Group

Fetch commands

## Syntax

FETCh:P25:CONStE:TRACe?

## Returns

#<num\_digit><num\_byte><l(1)><Q(1)><l(2)><Q(2)>...<l(n)><Q(n)>

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<l(n)> and <Q(n)> are the normalized I- and Q- coordinate values at the  $n^{\text{th}}$  data point. 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

FETCh:P25:CONStE:TRACe? might return #43848xxxx... (3848-byte data) for the P25 Constellation trace data.

## FETCh:P25:EDIagram:FDEVIation? (Query Only)

Returns the Frequency Deviation vs Time trace data with the X value.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Fetch commands

## Syntax

FETCh:P25:EDIagram:FDEVIation?

## Returns

#<num\_digit><num\_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> are the frequency deviation in Hz and time (symbols) coordinate pair at the  $n^{\text{th}}$  data point. 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

FETCh:P25:EDIAGRAM:FDEVIATION? might return #3160xxxx... (160-byte data) for the Frequency Deviation vs Time trace.

## FETCh:P25:EDlagram:FERRor? (Query Only)

Returns the frequency error in the P25 Eye Diagram display.

When Frequency Error is in Auto mode, this query returns the frequency error value. When in Manual mode, it returns the frequency offset.

### Conditions

Measurement view: P25 Eye Diagram

### Group

Fetch commands

### Syntax

```
FETCh:P25:EDIagram:FERRor?
```

### Returns

<freq\_error>::<NRf> is the frequency error in Hz.

### Examples

FETCh:P25:EDIAGRAM:FERROR? might return -10.7E+3, indicating the frequency Error is -10.7 kHz.

## FETCh:P25:EDlagram:I? (Query Only)

Returns the I versus Time trace data in the P25 Eye Diagram display.

### Conditions

Measurement view: P25 Eye Diagram

### Group

Fetch commands

### Syntax

```
FETCh:P25:EDIagram:I?
```

### Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> are the normalized I- and time (symbols) coordinate values at the  $n^{th}$  data point. 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

`FETCH:P25:EDIAGRAM:I?` might return #3160xxxx... (160-byte data) for the I versus Time trace.

## **FETCh:P25:EDlagram:Q? (Query Only)**

Returns the Q versus Time trace data in the P25 Eye Diagram display.

### Conditions

Measurement view: P25 Eye Diagram

### Group

Fetch commands

### Syntax

`FETCh:P25:EDIagram:Q?`

### Returns

#<num\_digit><num\_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> are the normalized Q- and time (symbols) coordinate values at the *n*th data point. 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

`FETCH:P25:EDIAGRAM:Q?` might return #3160xxxx... (160-byte data) for the Q Versus Time trace.

## **FETCh:P25:PVTime:BURSt:WIDTh? (Query Only)**

Returns the value of the measured time width of the burst packet from poweron to powerdown (or the end of waveform, if that occurs before powerdown), in seconds, for the P25 Power vs. Time trace.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Fetch commands

### Syntax

`FETCh:P25:PVTime:BURSt:WIDTh?`



## Returns

Floating point number that represents the value of the measured time width of the burst packet from power on to power down (or the end of waveform, if that occurs before power down), in seconds.

## Examples

`FETCH:P25:PVTIME:BURST:WIDTH?` might return `156.0000000000E-6`, which represents a width of the burst packet of 156  $\mu$ s.

## **FETCh:P25:PVTime:FALL:TIME:START? (Query Only)**

Returns the value of the start time of the falling edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:FALL:TIME:START?`

## Related Commands

`FETCh:P25:PVTime:FALL:TIME:STOP?`

## Returns

Floating point number which represents the value of the start time of the falling edge of the burst, in seconds.

## Examples

`FETCH:P25:PVTIME:FALL:TIME:START?` might return `6e-6`, which indicates that the start time of the falling edge is at 6  $\mu$ s.

## **FETCh:P25:PVTime:FALL:TIME:STOP? (Query Only)**

Returns the value of the stop time of the falling edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:FALL:TIME:STOP?`

## Related Commands

`FETCh:P25:PVTIME:FALL:TIME:START?`

## Returns

Floating point number which represents the value of the stop time of the falling edge of the burst, in seconds.

## Examples

`FETCH:P25:PVTIME:FALL:TIME:STOP?` might return  $7e-6$ , which indicates the stop time of the falling edge is at  $7\ \mu\text{s}$ .

## **FETCh:P25:PVTime:RISE:TIME? (Query Only)**

Returns the value of the width of the rising edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:RISE:TIME?`

## Related Commands

`FETCh:P25:PVTime:RTIME?`

## Returns

Floating point number which represents the value of the start time of the rising edge of the burst, in seconds.

## Examples

`FETCH:P25:PVTIME:RISE:TIME?` might return  $6e-6$ , which indicates the start time of the rising edge is  $6\ \mu\text{s}$ .

## **FETCh:P25:PVTime:RISE:TIME:START? (Query Only)**

Returns the value of the start time of the rising edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:RISE:TIME:START?`

## Related Commands

`FETCh:P25:PVTime:RISE:TIME?`

---

```
FETCh:P25:PVTime:RTIME?
```

### Returns

Floating point number which represents the value of the start time of the rising edge of the burst, in seconds.

### Examples

`FETCH:P25:PVTIME:RISE:TIME:START?` might return 6e-6, which indicates the start time of the rising edge is at 6  $\mu$ s.

## **FETCh:P25:PVTime:RISE:TIME:STOP? (Query Only)**

Returns the value of the stop time of the rising edge, in seconds.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Fetch commands

### Syntax

```
FETCh:P25:PVTime:RISE:TIME:STOP?
```

### Related Commands

```
FETCh:P25:PVTime:RISE:TIME:START?
```

```
FETCh:P25:PVTime:RTIME?
```

```
FETCh:P25:PVTime:RISE:TIME?
```

### Returns

Floating point number which represents the value of the stop time of the rising edge of the burst, in seconds.

### Examples

`FETCH:P25:PVTIME:RISE:TIME:STOP?` might return 7e-6, which indicates the width of the rising edge is 7  $\mu$ s.

## **FETCh:P25:PVTime:RTIME? (Query Only)**

Returns the value of the width of the rising edge, in seconds.

### Conditions

Measurement view: P25 Power vs. Time

### Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:RTIME?`

## Related Commands

`FETCh:P25:PVTime:RISE:TIME?`

## Returns

Floating point number which represents the value of the width of the rising edge of the burst, in seconds.

## Examples

`FETCh:P25:PVTIME:RTIME?` might return `7e-6`, which indicates the width of the rising edge is 7  $\mu$ s.

## FETCh:P25:PVTime:TRACe[:Y]? (Query Only)

Returns the vertical values (power) for the P25 Power vs. Time trace.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Fetch commands

## Syntax

`FETCh:P25:PVTime:TRACe[:Y]?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:P25:PVTIME:TRACE[:Y]?` might return `#43204` (3204-byte data), which represents the vertical values (power) for the P25 Power vs. Time trace.

## FETCh:P25:SUMMArY:LIMIt: SRA (Query Only)

Returns Limit of Symbol Rate Accuracy measurement results.

## Conditions

Measurement views: P25 Summary

---

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:LIMIT: SRA
```

## Arguments

None

## Returns

might return PASS, FAIL or NA.

## Examples

```
FETCH:P25:SUMMARY:LIMIT: SRA
```

## FETCh:P25:SUMMArY:MODULation:MEASurement:FIDelity (Query Only)

Returns the Modulation Fidelity measurement result.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:MODULation:MEASurement:FIDelity
```

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY
```

## FETCh:P25:SUMMArY:MODULation:MEASurement:FIDelity:STATus (Query Only)

Returns the Modulation Fidelity measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMary:MODUlation:MEASurement:FIDelity:STATus
```

## Arguments

None

## Returns

PASS means the result passed.

FAIL means the result failed.

NA means there is nothing to return.

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY:STATUS` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMary:MODUlation:MEASurement:FREQuency:.DEVIation:POSPeak**

Returns the Frequency Deviation, positive peak measurement results.

## Conditions

Measurement views: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQuency:.DEVIation:POSPeak
```

## Arguments

<integer>,<integer>

## Returns

Frequency Deviation, positive peak, result

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:.DEVIATION:POSPEAK <-2,-6>` indicates the coordinate value of the Frequency Deviation positive peak.

---

## **FETCh:P25:SUMMary:MODUlation:MEASurement:FREQuency:.DEVlation:POSPea k:STA**

Returns the status of Frequency Deviation, positive peak status.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Fetch commands

### **Syntax**

```
FETCh:P25:SUMMary:MODUlation:MEASurement:FREQuency:.DEVlation:POSPeak:STATus
```

### **Arguments**

None

### **Examples**

FETCh:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:.DEVIATION:POSPEAK:STATUS might return PASS, FAIL or NA.

## **FETCH:P25:SUMMary:MODUlation:MEASurement:FREQuency:ACCUracy (Query Only)**

Returns the Operational Frequency Accuracy measurement results.

### **Conditions**

Measurement views: P25 Summary

### **Group**

Fetch commands

### **Syntax**

```
FETCH:P25:SUMMary:MODUlation:MEASurement:FREQuency:ACCUracy
```

### **Arguments**

None

### **Returns**

Frequency Accuracy measurement result

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY`

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY (Query Only)**

Returns the Operational Frequency Accuracy measurement results in the P25 Summary display.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY?`

## Arguments

None

## Returns

<NRf> Returns Operational Frequency Accuracy

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY` might return 25.62, indicating the Operational Frequency Accuracy is 25.62 ppm.

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY:STATUS (Query Only)**

Returns the status of Frequency Accuracy measurement's result.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY: STATUS`



---

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY:STATUS` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK:STATUS (Query Only)**

Queries the status of Frequency Deviation, negative peak result in the P25 Summary display.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY: DEVIATION: NEGPEAK:STATUS?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION: NEGPEAK:STATUS` might return Pass, indicating that the Frequency Deviation for negative peak test result is Pass.

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK (Query Only)**

Returns the Frequency Deviation, negative peak, measurement results.

## Conditions

Measurement views: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY: DEVIATION:NEGPEAK
```

## Arguments

None

## Returns

Frequency Deviation measurement result

## Examples

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY: DEVIATION:NEGPEAK
```

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak : STATUS (Query Only)**

Queries the status of Frequency Deviation, positive peak result in the P25 Summary display.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY: DEVIATION:POSPeak: STATUS?
```

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak: STATUS might return Pass, indicating that the Frequency Deviation for positive peak test result is Pass.
```

---

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak (Query Only)**

Returns the Frequency Deviation, positive peak measurement results in the P25 Summary display.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Fetch commands

### **Syntax**

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY: DEVIATION:POSPeak?
```

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak might return 125.63, indicating the Frequency Deviation positive peak is 125.63 Hz.

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR (Query Only)**

Returns the Frequency Error measurement results.

### **Conditions**

Measurement views: P25 Summary

### **Group**

Fetch commands

### **Syntax**

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR
```

### **Arguments**

None

### **Returns**

Frequency Error measurement result

## **FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:SRA? (Query Only)**

Returns the Symbol Rate Accuracy measurement results.

### **Conditions**

Measurement views: P25 Summary

### **Group**

Fetch commands

### **Syntax**

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:SRA?
```

### **Arguments**

None

### **Returns**

Symbol Rate Accuracy measurement result

### **Examples**

```
FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:SRA?
```

## **FETCh:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUs (Query Only)**

Returns Status of Symbol Rate Accuracy measurement status.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Fetch commands

### **Syntax**

```
FETCh:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUs
```

### **Arguments**

None

### **Returns**

PASS, FAIL or NA.

## Examples

`FETCH:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUS` might return `PASS`, indicating that the status is passed.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF? (Query Only)**

Returns the HCPM Pmax-off measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF?`

## Returns

HCPM Pmax-off measurement results.

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF?` would return the HCPM Pmax-off measurement result.

## **FETCh:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUs? (Query**

Returns HCPM Pmax-off measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUs?`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUS?` might return `PASS`, `FAIL` or `NA`.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON? (Query Only)**

Returns the HCPM Pmax-on measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Fetch commands

### **Syntax**

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON?`

### **Returns**

HCPM Pmax-on measurement results.

### **Examples**

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON?` would return HCPM Pmax-on measurement results.

## **FETCh:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUs? (Query Only)**

Returns Status of HCPM Pmax-on measurement status.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Fetch commands

### **Syntax**

`FETCh:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUs?`

### **Arguments**

None

### **Examples**

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUS?` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS? (Query Only)**

Returns the HCPM Pss-max measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS?
```

## Returns

HCPM Pss-max measurement results.

## Examples

FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS? would return the HCPM Pss-max measurement result.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS (Query Only)**

Returns HCPM Pss-max measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS?
```

## Arguments

None

## Examples

FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS? might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM? (Query Only)**

Returns the HCPM Pss-min measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM?`

## Returns

HCPM Pss-min measurement results.

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM?` would return the HCPM Pss-min measurement result.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS? (Query Only)**

Returns status of HCPM Pss-min measurement status based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS?`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS?` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT? (Query Only)**

Returns the HCPM Offslot power results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands



## Syntax

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT?
```

## Returns

HCPM Offslot power results.

## Examples

FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT? would return the HCPM Offslot power result.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT:STATUS? (Query Only)**

Returns Status of HCPM Offslot measurement based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT:STATUS?
```

## Arguments

None

## Examples

FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT:STATUS? might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI (Query Only)**

Returns P\_ACP\_HI value which is displayed on P25 Summary display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI?
```

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI?` might return `-37.963` indicating the `P_ACP_HI` value is `-37.96` dBm.

## **FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:LOW (Query**

Returns `P_ACP_LOW` value which is displayed on P25 Summary display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:LOW`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:LOW?` might return `-40.125` indicating `P_ACP_LOW` value is `-40.13` dBm.

## **FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN (Query**

Returns Min Pk ACPR value which is displayed on P25 Summary Display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN`

---

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN?` might return 47.96 indicating Min Pk ACPR value is 47.96 dBm

## **FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN:STATus**

Returns status of Min Pk ACPR value based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:POWEr:MEASurement:HCPM:PEAK:ACPR:MIN:STATus`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN:STATUS?` might return PASS, FAIL or NA.

## **FETCh:P25:SUMMArY:POWEr:MEASurement:RF (Query Only)**

Returns RF measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:POWEr:MEASurement:RF`

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:POWER:MEASUREMENT:RF
```

## **FETCh:P25:SUMMArY:POWEr:MEASurement:RF:STATus (Query Only)**

Returns Status of RF measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:POWEr:MEASurement:RF:STATus
```

## Arguments

None

## Returns

PASS, FAIL, or NA.

## Examples

`FETCH:P25:SUMMARY:POWER:MEASUREMENT:RF:STATUS` might return PASS, indicating that PASS is the status result of the measurement.

## **FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:.ACQ:COUNT (Query Only)**

Returns HCPM t\_error\_1 acquisition count.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:.ACQ:COUNT?
```

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:.ACQ:COUNT?
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:ACQ:COUNT (Query Only)**

Returns the HCPM `t_error_1` acquisition count in P25 Summary display.

This is applicable when phase 2 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment: TERRor:ONE:ACQ:COUNT?
```

## Arguments

None

## Returns

<NR1> Range is 0 to 5

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:ACQ:COUNT` might return 1, indicating HCPM `t_error_1` acquisition count is 1.

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:ZERO:STATus (Query Only)**

Queries the status of the HCPM Time alignment's `t_error_0` measurement results in the P25 Summary display .

This is applicable when phase 2 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:ZERO:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:STATUS` might return Pass, indicating that the HCPM Time alignment's t\_error\_0 measurement test result is Pass.

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TOBSync:ACQ:COUNT (Query Only)**

Returns the HCPM Time alignment's tob\_sync acquisition count in P25 Summary display.

This is applicable when phase 2 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TOBSync:ACQ:COUNT?`

## Arguments

None

## Returns

<NR1> Range is 0 to 5

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSYNC:ACQ:COUNT` might return 2, indicating HCPM Time alignment's tob\_sync acquisition count is 2.

---

## **FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:STATus (Query Only)**

Returns the Status of HCPM Time alignment's t\_error\_1 measurement results in P25 Summary display.

This is applicable when phase 2 is selected as the standard in the Modulation Params tab.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Fetch commands

### **Syntax**

`FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ONE:STATus?`

### **Arguments**

None

### **Returns**

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

### **Examples**

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:STATUS` might return Pass, indicating that the HCPM Time alignment's t\_error\_1 measurement test result is Pass.

## **FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ZERO:ACQ:COUNT (Query Only)**

Returns the HCPM t\_error\_0 acquisition count in P25 Summary display.

This is applicable when phase 2 is selected as the standard in the Modulation Params tab.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Fetch commands

### **Syntax**

`FETCh:P25:SUMMArY:TRIGger:MEASurement:HCPM:TIME:ALIGNment:TERRor:ZERO:ACQ:COUNT?`

## Arguments

None

## Returns

<NR1> Range is 0 to 5

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT` might return 3, indicating HCPM `t_error_0` acquisition count is 3.

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:.Z ERO:S(Query Only)**

Returns status of HCPM Time alignment's `t_error_0` measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:.ZERO:STATus?`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:.ZERO:STATUS?` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMArY:TRIGGer:MEASurement:HCPM:TIME:ALIGNment:TERRor:O NE:.ACQ:COUNT?(Query Only)**

Returns the HCPM Time alignment's `t_error_1` acquisition count over which the `t_error_1` average is computed.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands



## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:.ACQ:COUNT?
```

## Returns

HCPM Time alignment's t\_error\_1 acquisition count.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:.ACQ:COUNT?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:?(Query Only)**

Returns the HCPM Time alignment's t\_error\_1 measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?
```

## Returns

HCPM Time alignment's t\_error\_1 measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO?**

Returns the HCPM Time alignment's t\_error\_0 measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO?
```

## Returns

HCPM Time alignment's t\_error\_0 measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT: TERROR:ZERO?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync? (Query Only)**

Returns the HCPM Time alignment's t\_obsync measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT: TOBSync?
```

## Returns

HCPM Time alignment's t\_obsync measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT: TOBSYNC?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:.ENCODER? (Query Only)**

Returns the Average Transmitter Encoder Attack Time measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:.ENCODER?
```

## Returns

Average Transmitter Encoder Attack Time measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:.ENCODER?
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:.ENCOrder:BI:STAT(Query Only)**

Returns status of Average Transmitter Power Encoder Attack Time, busy/idle, measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:.ENCOrder:BI:STATus?
```

## Arguments

None

## Returns

Average Transmitter Encoder Attack Time , busy/idle, measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:.ENCODER:BI:STATUS? might return  
PASS, FAIL or NA.
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:ENCOrder:STATus (Query Only)**

Returns the Status of Average Transmitter Encoder Attack Time measurement results in P25 Summary display.

This is applicable when phase 1 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:ENCOrder:STATus?
```

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS` might return Pass, indicating that Average Transmitter Encoder Attack Time measurement test result is Pass.

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:BI (Query Only)**

Returns Average Transmitter Power Attack Time, busy/idle measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:BI`

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI`

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:ACQ:COUN T(Query Only)**

Returns Average Transmitter Power Attack acquisition count.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASel:ATTAck:TIME: ACQ:COUNT
```

## Arguments

None

## Examples

FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT might return PASS, FAIL or NA.

## **FETCh:P25:SUMMary:TRIGger:MEASurement:PHASel:ATTAck:TIME:ACQ:COUNT:BI(Query Only)**

Returns the Average Transmitter Power Attack, busy/idle count in the P25 Summary display.

This is applicable when phase 1 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMary:TRIGger:MEASurement:PHASel:ATTAck:TIME: ACQ:COUNT:BI?
```

## Arguments

None

## Returns

<NR1>Range is 0 to 10

## Examples

FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI might return 8, indicating Average Transmitter Power Attack,busy/idle count is 8.

## **FETCh:P25:SUMMary:TRIGger:MEASurement:PHASel:ATTAck:TIME:ENCOrder:BI:STATus (Query Only)**

Returns the Status of Average Transmitter Encoder Attack Time (Busy/Idle) measurement results in P25 Summary display.

This is applicable when phase 1 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTAck:TIME: ENCOder: BI:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER: BI:STATUS` might return Pass, indicating that the Average Transmitter Encoder Attack Time (Busy/Idle) measurement test result was Pass.

## **FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTAck:TIME:ENCOder:STATus (Query Only)**

Returns the Status of Average Transmitter Encoder Attack Time measurement results in P25 Summary display.

This is applicable when phase 1 is selected as the standard in the Modulation Params tab.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

`FETCh:P25:SUMMary:TRIGger:MEASurement:PHASe1:ATTAck:TIME:ENCOder:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS` might return Pass, indicating that Average Transmitter Encoder Attack Time measurement test result is Pass.

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER? (Query Only)**

Returns the Average Transmitter Encoder Attack Time measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Fetch commands

### Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME: ENCODER?
```

### Returns

Average Transmitter Encoder Attack Time measurement results.

### Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME: ENCODER?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI ?(Query Only)**

Returns the Average Transmitter Encoder Attack Time, busy/idle, measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Fetch commands

### Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME: ENCODER:BI?
```

## Returns

Average Transmitter Encoder Attack Time , busy/idle, measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME: ENCODER:BI?
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWEr (Query Only)**

Returns Average Transmitter Power Attack Time measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWEr
```

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWEr:BI (Query Only)**

Returns Average Transmitter Power Attack Time, busy/idle measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWEr:BI
```



## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:BI:STATus(Query Only)**

Returns status of Average Transmitter Power Attack Time, busy/idle, measurement results based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME: POWER:BI:STATus
```

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI:STATUS` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:BI?(Query Only)**

Returns the Average Transmitter Power Attack Time, busy/idle, measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME: POWER:BI?
```

## Returns

Average Transmitter Power Attack Time, busy/idle, measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI?
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:STAtus(Query Only)**

Returns status of Average Transmitter Power Attack Time measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER:STAtus
```

## Arguments

None

## Examples

FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:STATUS might return PASS, FAIL or NA.

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER?(Query Only)**

Returns the Average Transmitter Power Attack Time measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:ATTAck:TIME:POWER?
```

## Returns

Average Transmitter Power Attack Time measurement results.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME: POWER?
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:THRUpUt:DELAY (Query Only)**

Returns Throughput delay measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:THRUpUt:DELAY
```

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY
```

## **FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:THRUpUt:DELAY:ACQ:COU NT (Query Only)**

Returns Throughput delay acquisition counts.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCh:P25:SUMMArY:TRIGGer:MEASurement:PHASe1:THRUpUt:DELAY:ACQ:COUNT
```

## Arguments

None

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY:ACQ:COUNT
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY:ACQ:COUNT?(Query Only)**

Returns the Throughput delay acquisition counts over which the Throughput delay average is computed.

## Conditions

Measurement view: P25 Summary

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY: ACQ:COUNT?
```

## Returns

Throughput delay acquisition counts.

## Examples

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY: ACQ:COUNT?
```

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY:STATUS (Query Only)**

Returns status of Throughput delay measurement results based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Fetch commands

## Syntax

```
FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY: STATUS
```

## Arguments

None

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY:STATUS` might return PASS, FAIL or NA.

## **FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY? (Query Only)**

Returns the Throughput delay measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Fetch commands

### Syntax

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY?`

### Returns

Throughput delay measurement results.

## Examples

`FETCH:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUPUT:DELAY?`

## **FETCH:PERROR[:MEASVIEW<y>]:FERROR? (Query only)**

Returns the frequency error in the Phase error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Phase error versus Time

MEASVIEW keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### Group

Fetch commands

### Syntax

`FETCH:PERROR[:MEASVIEW<y>]:FERROR?`

### Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`FETCh:PERRor:MEASview2:FERRor?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz for measurement view2.

## FETCh:PERRor[:MEASview<y>]:PEAK? (Query only)

Returns the peak value in the Phase error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Fetch commands

## Syntax

`FETCh:PERRor[:MEASview<y>]:PEAK?`

## Related Commands

[\*FETCh:PERRor\[:MEASview<y>\]:PINDEx?\*](#)

## Arguments

None

## Returns

`<peak>::<NRf>` is the peak phase error in percent (%).

## Examples

`FETCh:PERRor:MEASview2:PEAK?` might return `1.57`, indicating the peak phase error value is 1.57% for measurement view 2.

## FETCh:PERRor[:MEASview<y>]:PINDEx? (Query only)

Returns the time at the phase error peak.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:PERRor[:MEASview<y>]:PINdex?
```

## Related Commands

[FETCh:PERRor\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

<peak\_time>::<Nrf> is the time at the phase error peak in symbol number.

The unit can be changed by [SENSE]:DDEMod:TIME:UNITs command.

## Examples

FETCh:PERRor:MEASview2:PINdex? might return 68.000, indicating that the Phase error peak is at symbol #68 for measurement view 2.

## FETCh:PERRor[:MEASview<y>]:RMS? (Query only)

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:PERRor[:MEASview<y>]:RMS?
```

## Arguments

None

## Returns

`<rms>::<Nrf>` is the RMS phase error in percent (%).

## Examples

`FETCh:PERRor:MEASview2:RMS?` might return 0.382, indicating the RMS Phase error value is 0.382% for measurement view 2.

## FETCh:PERRor[:MEASview<y>]:TRACe? (Query only)

Returns the Phase error versus Time trace data.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Fetch commands

## Syntax

```
FETCh:PERRor[:MEASview<y>]:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the phase error in degrees at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:PERRor:MEASview2:TRACe?` might return `#42036xxxx...` (2036-byte data) for the Phase error versus Time Measview 2.

## FETCh:PHVTime[:MEASview<y>]? (Query Only)

Returns the Phase versus Time trace data.



## Conditions

Measurement views: Phase versus Time

## Group

Fetch commands

## Syntax

```
FETCh:PHVTime[:MEASview<y>]?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the phase in degrees at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:PHVTIME:MEASview1? might return #3160xxx... (160-byte data) for the Phase versus Time trace.

## FETCh:PHVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Fetch commands

## Syntax

```
FETCh:PHVTime[:MEASview<y>]:MAXimum?
```

## Related Commands

```
FETCh:PHVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

`<max>::<NRf>` is the maximum phase in degrees.

## Examples

`FETCH:PHVTIME:MEASview1:MAXIMUM?` might return 153.8, indicating the maximum phase is 153.8 °.

## FETCh:PHVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the phase is maximum.

## Conditions

Measurement views: Phase versus Time

## Group

Fetch commands

## Syntax

`FETCh:PHVTime[:MEASview<y>]:MAXLocation?`

## Related Commands

`FETCh:PHVTime[:MEASview<y>]:MAXimum?`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value 1.

## Returns

`<max_time>::<NRf>` is the time in seconds at which the phase is maximum.

## Examples

`FETCH:PHVTIME:MEASview1:MAXLOCATION?` might return 175.3E-9, indicating the I or Q level is maximum at 175.3 ns.

## FETCh:PHVTime[:MEASview<y>]:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Fetch commands

## Syntax

```
FETCh:PHVTime[:MEASview<y>]:MINimum?
```

## Related Commands

```
FETCh:PHVTime[:MEASview<y>]:MINLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min>::<NRf> is the minimum phase in degrees.

## Examples

```
FETCH:PHVTIME:MEASview1:MINIMUM? might return -176.3, indicating the minimum phase is -176.3 °.
```

## FETCh:PHVTime[:MEASview<y>]:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

## Conditions

Measurement views: Phase versus Time

## Group

Fetch commands

## Syntax

```
FETCh:PHVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
FETCh:PHVTime[:MEASview<y>]:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<NRf> is the time in seconds at which the phase is minimum.

## Examples

```
FETCH:PHVTIME:MEASview1:MINLOCATION? might return 450.7E-9, indicating the phase is minimum at 450.7 ns.
```

## **FETCh:PHVTime[:MEASview<y>]:RESult? (Query Only)**

Returns the Phase versus Time measurement results.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PHVTime[:MEASview<y>]:RESult?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Returns**

```
<max>,<max_time>,<min>,<min_time>
```

Where

<max>::<NRf> is the maximum phase in degrees.

<max\_time>::<NRf> is the time in seconds at which the phase is maximum.

<min>::<NRf> is the minimum phase in degrees.

<min\_time>::<NRf> is the time in seconds at which the phase is minimum.

### **Examples**

FETCh:PHVTIME::MEASview1:RESULT? might return 153.8,175.3E-9,-176.3,450.7E-9, indicating the maximum phase is 153.8 ° at 175.3 ns and the minimum phase is -176.3 ° at 450.7 ns.

## **FETCh:PM:PNEGative? (Query Only)**

Returns the negative peak phase deviation (-Pk) in the PM measurement.

### **Conditions**

Measurement views: PM

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PM:PNEGative?
```

## Arguments

None

## Returns

<-Pk> :: <Nrf> is the negative peak phase deviation in degrees.

## Examples

FETCH:PM:PNEGATIVE? might return -23.42, indicating the positive peak phase deviation is -23.42 °.

## FETCh:PM:PPOSitive? (Query Only)

Returns the positive peak phase deviation (+Pk) in the PM measurement.

## Conditions

Measurement views: PM

## Group

Fetch commands

## Syntax

FETCh:PM:PPOSitive?

## Arguments

None

## Returns

<+Pk> :: <Nrf> is the positive peak phase deviation in degrees.

## Examples

FETCH:PM:PPOSITIVE? might return 26.87, indicating the positive peak phase deviation is 26.87 °.

## FETCh:PM:PTPeak? (Query Only)

Returns the peak-peak phase deviation (Pk-Pk) in the PM measurement.

## Conditions

Measurement views: PM

## Group

Fetch commands

## Syntax

FETCh:PM:PTPeak?

## Arguments

None

## Returns

<Pk-Pk> : : <NRf> is the peak-peak phase deviation in degrees.

## Examples

FETCH:PM:PTPEAK? might return 46.34, indicating the peak-peak phase deviation is 46.34 °.

## FETCh:PM:RESult? (Query Only)

Returns the PM measurement results.

## Conditions

Measurement views: PM

## Group

Fetch commands

## Syntax

FETCh:PM:RESult?

## Arguments

None

## Returns

<+Pk>, <-Pk>, <RMS>, <Pk-Pk>

Where

<+Pk> : : <NRf> is the positive peak phase deviation in degrees.

<-Pk> : : <NRf> is the negative peak phase deviation in degrees.

<RMS> : : <NRf> is the RMS phase deviation in degrees.

<Pk-Pk> : : <NRf> is the peak-peak phase deviation in degrees.

## Examples

FETCH:PM:RESULT? might return 22.89,-23.45,15.12,46.34.

## FETCh:PM:RMS? (Query Only)

Returns the RMS phase deviation in the PM measurement.

## Conditions

Measurement views: PM

## Group

Fetch commands

## Syntax

FETCh:PM:RMS?

## Arguments

None

## Returns

<RMS>::<NRf> is the RMS phase deviation in degrees.

## Examples

FETCH:PM:RMS? might return 15.12, indicating the RMS frequency deviation is 15.12 °.

## FETCh:PNOise:ALL? (Query Only)

Returns all results of the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

FETCh:PNOise:ALL?

## Arguments

None

## Returns

<Cpower>,<Ferror>,<Pnoise>,<Tjitter>,<Rjitter>,<RFM>

Where

<Cpower>::<NRf> is the carrier power in dBm.

<Ferror>::<NRf> is the frequency error in Hz.

<Pnoise>::<NRf> is the RMS phase noise in degrees.

<Tjitter>::<NRf> is the total jitter in seconds.

<Rjitter>::<NRf> is the random jitter in seconds.

<RFM>::<NRf> is the residual FM in Hz.

## Examples

`FETCH:PNOISE:ALL?` might return `-9.455,1.235E+6,51.43,2.312E-9,4.178E-9,14.58`, indicating Carrier power: -9.455 dBm, Frequency error: 1.235 MHz, RMS phase noise: 51.43 °, Total jitter: 2.312 ns, Random jitter: 4.178 ns, and Residual FM: 14.58 Hz.

## **FETCh:PNOise:CARRier:FERRor? (Query Only)**

Returns the carrier frequency error in the phase noise measurement.

### Conditions

Measurement views: Phase noise

### Group

Fetch commands

### Syntax

`FETCh:PNOise:CARRier:FERRor?`

### Arguments

None

### Returns

<NRf> Carrier frequency error in Hz.

## Examples

`FETCH:PNOISE:CARRIER:FERROR?` might return `1.235E+6`, indicating that the carrier frequency error is 1.235 MHz.

## **FETCh:PNOise:CARRier:POWer? (Query Only)**

Returns the carrier power in the phase noise measurement.

### Conditions

Measurement views: Phase noise

### Group

Fetch commands

### Syntax

`FETCh:PNOise:CARRier:POWer?`

### Arguments

None



## Returns

<NRf> Carrier power in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:PNOise:CARRIER:POWER? might return -9.455, indicating that the carrier power is -9.455 dBm.

## FETCh:PNOise:JITTer? (Query Only)

Returns the jitter in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

FETCh:PNOise:JITTer?

## Arguments

None

## Returns

<NRf> Jitter in seconds.

## Examples

FETCh:PNOise:JITTER? might return 2.312E-9, indicating that the jitter is 2.312 ns.

## FETCh:PNOise:RESidual:FM? (Query Only)

Returns the residual FM in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

FETCh:PNOise:RESidual:FM?

## Arguments

None

## Returns

<NRf> Residual FM in Hz.

## Examples

FETCH:PNOISE:RESIDUAL:FM? might return 14.58, indicating that the residual FM is 14.58 Hz.

## FETCh:PNOise:RMS:PNOise? (Query Only)

Returns the RMS phase noise in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

FETCh:PNOise:RMS:PNOise?

## Arguments

None

## Returns

<NRf> RMS phase noise in degrees.

## Examples

FETCH:PNOISE:RMS:PNOISE? might return 51.43, indicating that the RMS phase noise is 51.43 °.

## FETCh:PNOise:SPECTrum<x>:X? (Query Only)

Returns the frequencies of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

```
FETCh:PNOise:SPECTrum<x>:X?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the frequency (Hz) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:PNOISE:SPECTRUM1:X? might return #516020xxxx... (16020-byte data) for the frequencies of Trace 1.

## FETCh:PNOise:SPECTrum<x>:XY? (Query Only)

Returns the frequency and phase noise pairs of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

```
FETCh:PNOise:SPECTrum<x>:XY?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the frequency (Hz) and phase noise (dBc/Hz) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:PNOISE:SPECTRUM1:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and phase noise pairs of the Trace 1.

## **FETCh:PNOise:SPECTrum<x>[:Y]? (Query Only)**

Returns the phase noise values of the specified trace.

The parameter `<x>` = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Fetch commands

## Syntax

`FETCh:PNOise:SPECTrum<x>[:Y]?`

## Arguments

None

## Returns

`#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<y(n)>` is the phase noise (dBc/Hz) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:PNOISE:SPECTRUM1:Y` might return `#516020xxxx...` (16020-byte data) for the phase noise values of Trace 1.

## **FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer? (Query Only)**

Returns the outliers value exceeding the upper limit in pulse cumulative histogram.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Number of outliers above higher limit

## Examples

FETCH:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:OUTLIER:HIGHER? might return 0.000, which is the value for the outliers that have exceeded the upper limit in the pulse cumulative histogram.

## FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer? (Query Only)

Returns the outliers value below the lower limit in pulse cumulative histogram.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Number of outliers below lower limit

## Examples

FETCH:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:OUTLIER:LOWER? might return 0.000, which is the value for the outliers below the lower limit in the pulse cumulative histogram.

## FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:COUNT? (Query Only)

Returns the number of pulses in the bin in the pulse cumulative histogram.

## Conditions

Measurement views: Pulse Cumulative Histogram

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:COUNT?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Number of pulses in the bin

## Examples

FETCH:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:PULSE:COUNT? might return 0 for the number of pulses in the bin.

## FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:TOTal? (Query Only)

Returns the total number of pulses in the pulse cumulative histogram.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:TOTal?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Total number of pulses

## Examples

`FETCH:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:PULSE:TOTAL?` might return 16 for total number pulses in the pulse cumulative histogram.

## FETCh:PULSe:CUMulative:STATistics[:MEASview<y>]

Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average and Standard Deviation values for the chosen measurement in the pulse cumulative statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Fetch commands

## Syntax

`FETCh:PULSe:CUMulative:STATistics[:MEASview<y>] <arg1>,<arg2>`

## Arguments

**arg1** can be one of the following measurements:

{AVERAge | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPPlE | RIPDb | DROOp | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFRequerror | MFRequerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude | IRTIME | PPPD | PPF | FABS}

**arg2** can be one of the following statistics results:

{TPCOunt|MAX|MAXTs|MIN|MINTs|PK2PK|AVG|SD|ALL}

## Returns

<puls\_cnt>|<max>|<max\_timstp>|<min>|<min\_timstp>|<pk\_pk>|<std\_dev>|< for one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq><rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev><phase\_dev> | <dlt\_a\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>

Where

<avg\_on>::<NRf> is the average power transmitted during pulse ON.

<pk\_pwr>::<NRf> is the Maximum power during pulse on.

<pk\_pwr>::<NRf> is the average power transmitted.

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB/-6 dB level (50%) of the user selected 100% level.

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level.

<fall>::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level.

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge.

<rep\_rate>::<NRf> is the inverse of repetition interval.

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage.

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period.

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top.

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB.

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time.

<droop\_db>::<NRf> is the Droop measurement expressed in dB.

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge.

<ovrshoot\_db>::<NRf> is the Over shoot measurement expressed in dB.

<p2p\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window.

<p2p\_freq>::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse.

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time.

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace.

<pp\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse.

<pp\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse.

<pk\_phase\_err>::<NRf> is the maximum phase error.

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point.

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time.

<phase\_dev>::<NRf> is the difference between the maximum and minimum Phase values measured during the ON time of a pulse.

<dlta\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency.

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe.

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe.

## Examples

FETCh:PULSe:CUMulative:STATistics:MEASview1 AVERage,MAX might return -1.3594852284, which is the max average value for the Average ON measurement.

## FETCh:PULSe:CUMulative:STATistics[:MEASview<y>] (Query Only)

Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the chosen measurement in the Pulse Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Fetch commands



## Syntax

```
FETCh:PULSe:CUMulative:STATistics[:MEASview<y>]? <arg1><arg2>
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<arg1> = {AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPple | RIPDb | DROOp | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFreqerror | MFReqerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude | IRTIME | PPPD | PPF | FABS}

<arg2> = {TPCOunt | MAX | MAXTs | MIN | MINTs | PK2PK | AVG | SD | ALL}

## Returns

<puls\_cnt> | <max> | <max\_timestep> | <min> | <min\_timestep> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcnt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlt\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcnt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlt\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe

## Examples

FETCh:PULSe:CUMulative:STATistics:MEASview1? AVERage,MAX might return -1.3594852284, indicating the max value for Average ON is -1.3594852284.

## FETCh:PULSe:OGRAM[:MEASview<y>]:MAX:FRAME:COUNT (Query Only)

Queries the frame count value in the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Fetch commands

## Syntax

FETCh:PULSe:OGRAM[:MEASview<y>]:MAX:FRAME:COUNT?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> Total number of fast frames

## Examples

FETCh:PULSe:OGRAM:MEASview1:MAX:FRAME:COUNT? might return 15, indicating the total number of fast frames in the acquisition of pulse-Ogram is 15.

## FETCh:PULSe:OGRAM[:MEASview<y>]:RBW (Query Only)

Queries the resolution bandwidth value only when Frequency vs Time display is opened in the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Fetch commands

### Syntax

```
FETCh:PULSe:OGRAM[:MEASview<y>]:RBW?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

<NRf>

### Examples

FETCh:PULSe:OGRAM:MEASview1:RBW? might return 222.9199886322E+3, indicating the RBW value in Pulse-Ogram is 222.9199886322E+3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:ADVanced (Query Only)

Queries the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the chosen measurement in the Pulse Table display.

### Conditions

Measurement views: Pulse Table

### Group

Fetch commands

### Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:ADVanced <arg1><arg2>
```

### Arguments

<arg1>={Pulse\_number}

<arg2>= {AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPPlE | RIPDb | DROOp | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFRequency | MFRReqerror | RMSPherror | MPHerror | FRDeViation | FDELta | PHDeViation | IRAMplitude | TIME | IRTIME | PPPD | PPF | FABS}

## Returns

<puls\_cnt> | <max> | <max\_tmstamp> | <min> | <min\_tmstamp> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlt\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<delta\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe <time>::<NRf> is the dtme in seconds relative to the time reference point in the first acquisition record in the data set

## Examples

FETCH:PULSE[:RESULT]:MEASview1:ADVANCED 1,AVERAge might return -1.4063608208, indicating the average ON value for the pulse number 1 is -1.4063608208.

## FETCh:PULSe[:RESult][[:MEASview<y>]:ADVanced:RANGe (Query only)

Queries the Pulse Count, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the selected measurement in the Pulse Table display.

## Conditions

Measurement views: Pulse Table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:ADVanced:RANGe <arg1><arg2><arg3>

## Arguments

<arg1>={Start\_Pulse\_number}

<arg2>={End\_Pulse\_number}

<arg3>= {AVERAge | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPple | RIPDb | DROop | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFReqerror | MFReqerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude|TIME | IRTime | PPPD| PPFd | FABS}

## Returns

<puls\_cnt> | <max> | <max\_timstamp> | <min> | <min\_timstamp> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <delta\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number.

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on.

<avg\_tx>:: <NRf> is the average power transmitted .

<width>:: <NRf> is the time from the rising edge to the falling edge at the -3 dB or -6 dB level (50%) of the user selected 100% level.

<rise>:: <NRf> is the time required for a signal to rise from 10% to 90% or (20% to 80%) of the user selected 100% level.

<fall> :: <NRf> is the time required for a signal to fall from 90% to 10% or (80% to 20%) of the user selected 100% level.

<rep\_int>:: <NRf> is the time from a pulse rising edge to the next pulse rising edge.

<rep\_rate> :: <NRf> is the inverse of repetition interval.

<duty\_factor\_pcmt>:: <NRf> is the ratio of the width to the pulse period, expressed as a percentage.

<duty\_fact\_ratio>:: <NRf> is the ratio of the pulse width to the pulse period.

<ripple>:: <NRf> is the Ripple is the peak-to-peak ripple on the pulse top.

<ripple\_db>:: <NRf> is the Ripple measurement expressed in dB.

<droop>:: <NRf> is the Droop is the power difference between the beginning and the end of the pulse on time.

<droop\_db>:: <NRf> is the Droop measurement expressed in dB.

<ovrshoot>:: <NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge.

<ovrshoot\_db>:: <NRf> is the Overshoot measurement expressed in dB.

<pp\_phase>:: <NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window.

<pp\_freq> :: <NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse.

<rms\_phase\_err>:: <NRf> is the RMS average of the Phase versus Time trace, computed over the Measurement Time.

<rms\_freq\_err>:: <NRf> is the RMS average of the Freq error versus Time trace.

<p2p\_phase>:: <NRf> is the frequency difference between the current pulse and the first pulse.

<p2p\_freq>:: <NRf> is the frequency difference between the current pulse and immediate previous pulse.

<pk\_phase\_err>:: <NRf> is the maximum phase error.

<abs\_freq>:: <NRf> is the absolute pulse frequency measured at a specific point.

<freq\_dev>:: <NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time.

<phase\_dev>:: <NRf> is the Phase deviation difference between the maximum and minimum Phase values measured during the on time of a pulse.

<dlta\_freq>:: <NRf> is the difference from the measurement frequency to each pulse frequency.

<imp\_resp\_amp>:: <NRf> is the difference in dB between the levels of the main lobe and highest side lobe.

<imp\_resp\_time>:: <NRf> is the difference in time between the main lobe and highest side lobe .

<time>:: <NRf> is the dtme in seconds relative to the time reference point in the first acquisition record in the data set.

## Example

```

FETCH:PULSE[:RESULT]:MEASview1:ADVANCED:RANGe 2048,2060,AVERAge might return
-3,23.8467659801E-3,23.8566566259E-3,23.8667335361E-3,23.8771531731E-3,23.886650800
7E-3,23.8966420293E-3,23.9066667855E-3,23.9170398563E-3,23.9367540926E-3,23.9467956

```

126E-3, 23.9567346871E-3, 23.9670332521E-3, 23.9866096526E-3, 2, indicating the average ON value for the pulse number 2048 to 2060.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:ATX? (Query Only)**

Returns the average transmitted power in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:ATX?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>,<ATX(1)>,< ATX(2)>,...<ATX(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ATX(n)>::<NRf> is the average transmitted power for the pulse with the number of [first\_pulse\_num + n - 1] in dBm.

The unit can be changed to watts by the [SENSE]:POWER:UNITs command. This measurement gives valid results only for power units ( Watts, dBm, dBuW, dBW).

### **Examples**

FETCh:PULSE:RESULT:MEASview1:ATX? might return 0,-18.57,-18.73,-18.20, -18.53 for Pulse 0 to 3.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:AVERage? (Query Only)**

Returns the average on power in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:AVERage?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,< avg (1)>,< avg (2)>,...<avg (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<avg (n)>::<NRf> is the average on power for the pulse with the number of [first\_pulse\_num + n - 1] in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

This measurement gives valid results only for power units ( Watts, dBm, dBuW, dBW).

## Examples

FETCh:PULSe:RESult:MEASview1:AVERAGE? might return 0,-2.354,-2.368,-2.343,-2.358 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:DROdb? (Query Only)

Returns the droop in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:DROdb?

## Related Commands

FETCh:PULSe[:RESult][[:MEASview<y>]:DROop?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,<droop (1)>,<droop (2)>,...<droop (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<droop (n)>::<NRf> is the droopdB for the pulse number [first\_pulse\_num + n - 1] in percent (%) of watts or volts.



## Examples

`FETCH:PULSE:RESULT:MEASview1:DRODB?` might return 1,-0.4,0.32,0.2,-0.1 for Pulse 1 to 4.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:DROop? (Query Only)**

Returns the droop in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

`FETCh:PULSe[:RESult][[:MEASview<y>]:DROop?`

### Related Commands

`FETCh:PULSe[:RESult][[:MEASview<y>]:DRODb?`

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<first\_pulse\_num>,<droop (1)>,<droop (2)>,...<droop (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<droop (n)>::<NRf> is the droop for the pulse number [first\_pulse\_num + n - 1] in percent (%) of watts or volts.

## Examples

`FETCH:PULSE:RESULT:MEASview1:DROOP?` might return 0,-270.9E-3,-193.0E-3,-242.7E-3,-177.5E-3 for Pulse 0 to 3.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:DUTPct? (Query Only)**

Returns the duty factor (%) in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:DUTPct?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<duty_pct(1)>,<duty_pct(2)>,...<duty_pct(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<duty\_pct(n)>::<NRf> is the duty factor for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%).

## Examples

FETCH:PULSE:RESULT:MEASview1:DUTPCT? might return 0,28.94,28.96,29.00,29.01 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:DUTRatio? (Query Only)

Returns the duty factor (ratio) in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:DUTRatio?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<duty_ratio(1)>,<duty_ratio(2)>,...<duty_ratio(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<duty\_ratio(n)>::<NRf> is the duty factor for the pulse with the number of [first\_pulse\_num + n - 1] (no unit).

## Examples

FETCH:PULSE:RESULT:MEASview1:DUTRATIO? might return 0,289.4E-3,289.6E-3, 290.0E-3,290.1E-3 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:FABS? (Query Only)

Returns the computed absolute frequency from the carrier in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:FABS?
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

```
<first_pulse_num>,<abs_freq(1)>,<abs_freq(2)>,. . . <abs_freq(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<abs\_freq(n)>::<NR1> is the absolute frequency for the pulse with the number of [first\_pulse\_num + n — 1] in Hz.

### Examples

FETCh:PULSE[:RESULT]:MEASview1:FABS? might return 1.000000000, 919.872640000E+6, 1.1664642560E+9, 1.1866586880E+9, 1.1867939840E+9, 1.1882243840E+9, 933.2595840000E+6, 1.1729175040E+9, 1.1818649600E+9, 1.1868037120E+9, 1.1832893440E+9, 933.4762880000E+6, 1.1805880320E+9, 1.1844794880E+9, 1.1914785280E+9, 1.1929450240E for is the number of the first pulse and the absolute frequency for each pulse.

## FETCh:PULSe[:RESult][[:MEASview<y>]:FALL? (Query Only)

Returns the fall time in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:FALL?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,<fall (1)>,<fall (2)>,...<fall (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<fall (n)>::<NRf> is the fall time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:FALL? might return 0,110.3E-9,90.45E-9, 95.03E-9,111.9E-9 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:FDELta? (Query Only)

Returns the computed delta frequency from the carrier in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:FDELta?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,<freq\_delta (1)>,<freq\_delta (2)>,...<freq\_delta (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<freq\_delta (n)>::<NRf> is the frequency delta for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

FETCH:PULSE:RESULT:MEASview1:FDELTA? might return 1.0000000000,163.6692962646,82.8298492432,21.8995475769,89.2963562012,4 for the first pulse and each pulse found.

## FETCh:PULSe[:RESult][[:MEASview<y>]:FRDeviation? (Query Only)

Returns the frequency deviation in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:FRDeviation?
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

```
<first_pulse_num>,<freq_dev(1)>,<freq_dev(2)>,...<freq_dev(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<freq\_dev(n)>::<NRf> is the frequency deviation for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

### Examples

FETCh:PULSE:RESULT:MEASview1:FRDEVIATION? might return 1,740.6E+3, 736.5E+3,718.3E+3,672.2E+3 for Pulse 1 to 4.

## FETCh:PULSe[:RESult][[:MEASview<y>]:IRAMplitude? (Query Only)

Returns the pulse impulse response amplitude measurement in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:IRAMplitude?
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <amplitude (1)>, <amplitude (2)>, ... <amplitude (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<amplitude (n)>::<NRf> is the sidelobe level, in dB, from the mainlobe for the pulse with the number of [first\_pulse\_num + n - 1].

## Examples

FETCH:PULSE:RESULT:MEASview1:IRAMPLITUDE? might return 1.0000000000,-45.3,-47.5,-50.2 for Pulse 1 through 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:IRTime? (Query Only)

Returns the impulse response time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:IRTime?

## Returns

<first\_pulse\_num>, <time (1)>, <time (2)>, . . <time (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<time (n)>::<NRf> is the sidelobe offset time from the mainlobe for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:IRTIME? might return  
1.0000000000,806.6708687693E-9,-846.6595318168E-9,-873.3477443457E-9 for Pulse 1 through 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:MFRreqerror? (Query Only)

Returns the maximum frequency error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:MFReqerror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<max_freq_err(1)>,<max_freq_err(2)>,...<max_freq_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<max\_freq\_err(n)>::<NRf> is the maximum frequency error for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

FETCH:PULSE:RESULT:MEASview1:MFREQERROR? might return 1,597.5E+3,675.8E+3,642.8E+3,598.2E+3 for Pulse 1 to 4.

## FETCh:PULSe[:RESult][:MEASview<y>]:MPHerror? (Query Only)

Returns the maximum phase error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:MPHerror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<max_phase_err(1)>,<max_phase_err(2)>,...<max_phase_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<max\_phase\_err(n)>::<NRf> is the maximum phase error for the pulse with the number of [first\_pulse\_num + n - 1] in degrees.

## Examples

FETCH:PULSE:RESULT:MEASview1:MPHERROR? might return 1,-9.221,-8.413,-11.853,-10.258 for Pulse 1 to 4.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:OVEDb? (Query Only)**

Returns the overshoot in dB in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:OVEDb?
```

### **Related Commands**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:OVERshoot?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>, <overshoot (1)>, <overshoot (2)>, ... <overshoot (n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<overshoot (n)>::<NRf> is the overshoot for the pulse with the number of [first\_pulse\_num + n - 1] in dB.

### **Examples**

FETCh:PULSE:RESULT:MEASview1:OVERDB? might return 1,1.2,2.3,1.0,0.2 for Pulse 1 to 4.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:OVERshoot? (Query Only)**

Returns the overshoot in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:OVERshoot?
```



## Related Commands

FETCh:PULSe[:RESult][[:MEASview<y>]:OVEDb?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <overshoot(1)>, <overshoot(2)>, ... <overshoot(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<overshoot(n)>::<NRf> is the overshoot for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%) of watts or volts.

## Examples

FETCH:PULSE:RESULT:MEASview1:OVERSHOOT? might return 1,1.2,2.3,1.0,0.2 for Pulse 1 to 4.

## FETCh:PULSe[:RESult][[:MEASview<y>]:PHDeviation? (Query Only)

Returns the phase deviation in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:PHDeviation?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <phase\_dev(1)>, <phase\_dev(2)>, ... <phase\_dev(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<phase\_dev(n)>::<NRf> is the phase deviation for the pulse with the number of [first\_pulse\_num + n - 1] in degrees.

## Examples

FETCH:PULSE:RESULT:MEASview1:PHDEVIATION? might return 1,11.658,9.640,10.509,8.272 for Pulse 1 to 4.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:PPFD? (Query Only)**

Returns the computed pulse to pulse frequency difference from the carrier in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:PPFD?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>,<ppfd(1)>,<ppfd(2)>,. . . <ppfd(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ppfd(n)>::<NRf> is the pulse to pulse frequency difference with the number of [first\_pulse\_num + n — 1] in Hz.

### **Examples**

FETCh:PULSE[:RESULT]:MEASview1:PPFD? might return 1.000000000,"-", 246.5916160000E+6, 20.1943520000E+6, 135.2960000000E+3, 1.4304160000E+6, -254.9647520000E+6, 239.6579520000E+6, 8.9474560000E+6, 4.9386560000E+6, -3.5143840000E+6, -249.8130240000E+6, 247.1117760000E+6, 3.8915200000E+6, 6.9989120000E+6, 1.4665920000E+6 is the number of the first pulse and the pulse frequency difference for each pulse.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:PPFRequency? (Query Only)**

Returns the pulse-pulse carrier frequency in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:PPFRequency?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <pp\_freq(1)>, <pp\_freq(2)>, ... <pp\_freq(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pp\_freq(n)>::<NRf> is the pulse-pulse carrier frequency for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

FETCH:PULSE:RESULT:MEASview1:PPFREQUENCY? might return 0,0.000,1.258E+3,-3.121E+3,1.862E+3 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:PPOWER? (Query Only)

Returns the peak power in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:PPOWER?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <pk\_power(1)>, <pk\_power(2)>, ... <pk\_power(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pk\_power(n)>::<NRf> is the peak power for the pulse with the number of [first\_pulse\_num + n - 1] in dBm.

The unit can be changed by the [SENSe]:POWER:UNITs command.

This measurement gives valid results only for power units( Watts, dBm, dBuW, dBW).

## Examples

FETCH:PULSE:RESULT:MEASview1:PPOWER? might return 0,-2.26,-2.27,-2.23,-2.25 for Pulse 0 to 3.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:PPPD? (Query Only)**

Returns the computed pulse to pulse phase difference from the carrier in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:PPPD?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>,<pppd(1)>,<pppd(2)>,. . .<pppd(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pppd(n)>::<NRf> is the pulse to pulse phase difference with the number of [first\_pulse\_num + n — 1] in Hz.

### **Examples**

FETCh:PULSE[:RESULT]:MEASview1:PPPD? might return 1.0000000000,"-", 96.2469482422, 24.9118175507, 5.8206896782, 16.7037734985, -151.8063964844, 100.7584228516,36.6761512756, -3.8614616394, 12.9028282166, -150.9820556641, 130.5854492188, 9.1259155273, -7.4177660942, 6.0613451004, -142.4224853516 is the number of the first pulse and the pulse to pulse phase difference for each pulse.

## **FETCh:PULSe[:RESult][[:MEASview<y>]:PPPHase? (Query Only)**

Returns the pulse-pulse carrier phase in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe[:RESult][[:MEASview<y>]:PPPHase?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <pp\_phase(1)>, <pp\_phase(2)>, ... <pp\_phase(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pp\_phase(n)>::<NRf> is the pulse-pulse carrier phase for the pulse with the number of [first\_pulse\_num + n - 1] in degrees.

## Examples

FETCH:PULSE:RESULT:MEASview1:PPPHASE? might return 0,0.000,21.66,46.76,57.56 for Pulse 0 to 3.

## FETCh:PULSe[:RESult]:MEASview<y>:RINTerval? (Query Only)

Returns the repetition interval in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCH:PULSe[:RESult]:MEASview<y>:RINTerval?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <rep\_int(1)>, <rep\_int(2)>, ... <rep\_int(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<rep\_int(n)>::<NRf> is the repetition interval for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:RINTERVAL? might return 0,16.03E-6,16.08E-6,16.07E-6,16.02E-6 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:RIPDb? (Query Only)

Returns the ripple in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:RIPDb?
```

## Related Commands

```
FETCh:PULSe[:RESult][:MEASview<y>]:RIPPlE?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<ripple(1)>,<ripple(2)>,...<ripple(n)>
```

Where

```
<first_pulse_num>::<NR1> is the first pulse number.
```

```
<ripple(n)>::<NRf> is the ripple for the pulse with the number of [first_pulse_num + n - 1] in dB.
```

## Examples

```
FETCH:PULSE:RESULT:MEASview1:RIPDB? might return 1,0.4,0.32,0.2,0.1 for Pulse 1 to 4.
```

## FETCh:PULSe[:RESult][:MEASview<y>]:RIPPlE? (Query Only)

Returns the ripple in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:RIPPlE?
```

## Related Commands

```
FETCh:PULSe[:RESult][:MEASview<y>]:RIPDb?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <ripple(1)>, <ripple(2)>, ... <ripple(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ripple(n)>::<NRf> is the volts or watts ripple for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%).

## Examples

FETCH:PULSE:RESULT:MEASview1:RIPPLE? might return 0,106.5E-3,177.6E-3, 148.3E-3,148.5E-3 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:RISE? (Query Only)

Returns the rise time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

FETCh:PULSe[:RESult][[:MEASview<y>]:RISE?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <rise(1)>, <rise(2)>, ... <rise(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<rise(n)>::<NRf> is the rise time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:RISE? might return 0,92.94E-9,115.9E-9, 115.1E-9,97.45E-9 for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:RMSFreqerror? (Query Only)

Returns the RMS frequency error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:RMSFreqerror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<RMS_freq_err(1)>,<RMS_freq_err(2)>,...<RMS_freq_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<RMS\_freq\_err(n)>::<NRf> is the RMS frequency error for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

FETCH:PULSE:RESULT:MEASview1:RMSFREQERROR? might return 1,51.54E+3, 69.20E+3,64.21E+3,51.02E+3 for Pulse 1 to 4.

## FETCh:PULSe[:RESult][[:MEASview<y>]:RMSPherror? (Query Only)

Returns the RMS phase error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][[:MEASview<y>]:RMSPherror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<RMS_phase_err(1)>,<RMS_phase_err(2)>,...<RMS_phase_err(n)>
```

Where



`<first_pulse_num>::<NR1>` is the first pulse number.

`<RMS_phase_err(n)>::<NRf>` is the RMS phase error for the pulse with the number of `[first_pulse_num + n - 1]` in degrees.

## Examples

`FETCH:PULSE:RESULT:MEASview1:RMSPHERROR?` might return `1,908.4E-3,752.8E-3,981.7E-3,886.4E-3` for Pulse 1 to 4.

## FETCh:PULSe[:RESult][[:MEASview<y>]:RRATe? (Query Only)

Returns the repetition rate in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

### Syntax

`FETCh:PULSe[:RESult][[:MEASview<y>]:RRATe?`

### Arguments

The parameter `<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<first_pulse_num>,<rep_rate(1)>,<rep_rate(2)>,...<rep_rate(n)>`

Where

`<first_pulse_num>::<NR1>` is the first pulse number.

`<rep_rate(n)>::<NRf>` is the repetition rate for the pulse with the number of `[first_pulse_num + n - 1]` in Hz.

## Examples

`FETCH:PULSE:RESULT:MEASview1:RRATE?` might return `0,62.50E+3,62.52E+3, 62.51E+3,62.49E+3` for Pulse 0 to 3.

## FETCh:PULSe[:RESult][[:MEASview<y>]:TIME? (Query Only)

Returns the time in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:TIME?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<time(1)>,<time(2)>,...<time(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<time(n)>::<NRf> is the time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:TIME? might return 1,7.937E-3,8.436E-3, 6.504E-3,9.876E-3 for Pulse 1 to 4.

## FETCh:PULSe[:RESult][:MEASview<y>]:WIDTh? (Query Only)

Returns the pulse width in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Fetch commands

## Syntax

```
FETCh:PULSe[:RESult][:MEASview<y>]:WIDTh?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<width(1)>,<width(2)>,...<width(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<width(n)>::<NRf> is the pulse width for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

FETCH:PULSE:RESULT:MEASview1:WIDTh? might return 0,4.630E-6,4.632E-6, 4.639E-6,4.642E-6 for Pulse 0 to 3.

## FETCH:PULSe:STATistics[:MEASview<y>]: ADVanced (Query Only)

Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in Pulse Statistics display.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

```
FETCH:PULSe:STATistics[:MEASview<y>]:ADVanced? <arg1>,<arg2>
```

### Arguments

<arg1>= {AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio |RIPPlE | RIPDb | DROOp | DRODb | OVERshoot | OVEDb | PPPHase | PPFRequency | RMSFreqerror | MFRreqerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude|TIME | IRTime | PPPD| PPF | FABS}

<arg2>={TPCOunt|MAX|MIN|PK2PK|AVG|SD|ALL}

### Returns

<puls\_cnt> | <max> | <min> | <pk\_pk> | <avg> | <std\_dev> | <all> For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlta\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB/-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::

<ovrshoot\_db>::

<pp\_phase>::

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::

<rms\_freq\_err>::

<p2p\_phase>::

<p2p\_freq>::

<pk\_phase\_err>::

<abs\_freq>::

<freq\_dev>::

<phase\_dev>::

<dlta\_freq>::

<imp\_resp\_amp>::

<imp\_resp\_time>::

## Examples

FETCH:PULSE:STATISTICS:MEASview1:ADVANCED RISE, MAX might return 7.9786177665E-9,"-", indicating the maximum rise value is 7.9786177665E-9.

## FETCh:PULSe:STATistics[:MEASview<y>]:ATX (Query Only)

Returns the average transmitted power in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:ATX?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<ATX\_avg>, <ATX\_min>, <ATX\_max>

Where

<ATX\_avg>::<NRf> is the average of the average transmitted power.

<ATX\_min>::<NRf> is the minimum of the average transmitted power.

<ATX\_max>::<NRf> is the maximum of the average transmitted power. Unit: dBm.

The unit can be changed to Watts by the [SENSE]:POWER:UNITs command. Volt is invalid in the average transmitted power measurement.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:ATX? might return -18.51,-18.74,-18.12 for the average transmitted power in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:AVERAge (Query Only)

Returns the average on power in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:AVERAge?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<avg\_avg>, <avg\_min>, <avg\_max>

Where

<avg\_avg>::<NRf> is the average of the average on power.

<avg\_min>::<NRf> is the minimum of the average on power.

<avg\_max>::<NRf> is the maximum of the average on power. Unit: dBm.

The unit can be changed to watts by the [SENSE]:POWER:UNITs command.

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:AVERAGE?` might return -2.35,-2.36,-2.34 for the average on power in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:DRoDb (Query Only)**

Returns the droop in dB in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:DRoDb?`

### Related Commands

`FETCh:PULSe:STATistics[:MEASview<y>]:DRoOp?`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<droop_avg>`, `<droop_min>`, `<droop_max>`

Where

`<droop_avg>`::`<NRf>` is the average droop.

`<droop_min>`::`<NRf>` is the minimum droop.

`<droop_max>`::`<NRf>` is the maximum droop. Unit: in dB.

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:DRoDb?` might return 22.67E-3,-613.5E-3, 633.8E-3 for the droop in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:DRoOp (Query Only)**

Returns the droop in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:DROop?
```

## Related Commands

```
FETCh:PULSe:STATistics[:MEASview<y>]:DRODb?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<droop_avg>, <droop_min>, <droop_max>
```

Where

<droop\_avg>::<NRf> is the average droop.

<droop\_min>::<NRf> is the minimum droop.

<droop\_max>::<NRf> is the maximum droop. Unit: Percent (%) by watts.

## Examples

FETCh:PULSe:STATISTICS:MEASVIEW2:DROop? might return 22.67E-3,-613.5E-3, 633.8E-3 for the droop in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:DUTPct (Query Only)

Returns the duty factor (%) in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:DUTPct?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<duty\_pct\_avg>,<duty\_pct\_min>,<duty\_pct\_max>

Where

<duty\_pct\_avg>::<NRf> is the average duty factor.

<duty\_pct\_min>::<NRf> is the minimum duty factor.

<duty\_pct\_max>::<NRf> is the maximum duty factor. Unit: Percent (%).

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:DUTPct? might return 2.437,2.310,2.657 for the duty factor (%) in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:DUTRatio (Query Only)

Returns the duty factor (ratio) in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:DUTRatio?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<duty\_ratio\_avg>,<duty\_ratio\_min>,<duty\_ratio\_max>

Where

<duty\_ratio\_avg>::<NRf> is the average duty factor.

<duty\_ratio\_min>::<NRf> is the minimum duty factor.

<duty\_ratio\_max>::<NRf> is the maximum duty factor. Unit: None.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:DUTRATIO? might return 24.37E-3,23.11E-3, 26.57E-3 for the duty factor (ratio) in the pulse statistics.



## **FETCh:PULSe:STATistics[:MEASview<y>]:FABS (Query Only)**

Returns the absolute frequency in the Pulse Statistics display.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe:STATistics[:MEASview<y>]:FABS?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf>

### **Examples**

FETCh:PULSe:STATISTICS:MEASVIEW2:FABS? might return 99.9846400000E+6, 99.9846400000E+6, 99.9846400000E+6, indicating the absolute frequency value.

## **FETCh:PULSe:STATistics[:MEASview<y>]:FALL (Query Only)**

Returns the fall time in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe:STATistics[:MEASview<y>]:FALL?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<fall\_avg>, <fall\_min>, <fall\_max>

Where

<fall\_avg>::<NRf> is the average fall time.

<fall\_min>::<NRf> is the minimum fall time.

<fall\_max>::<NRf> is the maximum fall time. Unit: Seconds.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:FALL? might return 70.27E-9,69.62E-9, 71.27E-9 for the fall time in the pulse statistics.

## FETCH:PULSe:STATistics[:MEASview<y>]:FDELta (Query Only)

Return the delta frequency in the pulse statistics.

### Group

Fetch commands

### Syntax

FETCH:PULSe:STATistics[:MEASview<y>]:FDELta?

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<freq\_delta\_avg>,<freq\_delta\_min>,<freq\_delta\_max>

Where

<freq\_delta\_avg>::<NRf> is the average frequency deviation.

<freq\_delta\_min>::<NRf> is the minimum frequency deviation.

<freq\_delta\_max>::<NRf> is the maximum frequency deviation. Unit: Hz.

## Examples

FETCH:PULSe:STATistics:MEASview1:FDELta? might return 77.9785690308,-110.1940994263,287.8957519531 for the frequency delta in the pulse statistics.

## FETCH:PULSe:STATistics[:MEASview<y>]:FRDeviation (Query Only)

Returns the frequency deviation in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

## Syntax

```
FETCH:PULSE:STATISTICS[:MEASview<y>]:FRDeviation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<freq_dev_avg>, <freq_dev_min>, <freq_dev_max>
```

Where

<freq\_dev\_avg>::<NRf> is the average frequency deviation.

<freq\_dev\_min>::<NRf> is the minimum frequency deviation.

<freq\_dev\_max>::<NRf> is the maximum frequency deviation. Unit: Hz.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:FRDEVIATION? might return 754.1E+3, 660.5E+3,835.2E+3 for the frequency deviation in the pulse statistics.

## FETCH:PULSE:STATISTICS[:MEASview<y>]:IRAmplitude (Query Only)

Returns the Pulse Impulse Response Amplitude measurement in the pulse statistics.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCH:PULSE:STATISTICS[:MEASview<y>]:IRAmplitude?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<amp;_avg>, <amp;_min>, <amp;_max>
```

Where

<amp;\_avg>::<NRf> is the average of the impulse response amplitude.

<amp;\_min>::<NRf> is the minimum of the impulse response amplitude.

<amp;\_max>::<NRf> is the maximum of impulse response amplitude. Unit: dB.

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:IRAMPLITUDE?` might return -45.3,-47.1,-43.2 for the impulse response amplitude in the pulse statistics.

## **FETCH:PULSe:STATistics[:MEASview<y>]:IRTime (Query Only)**

Returns the impulse response time in the pulse statistics.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCH:PULSe:STATistics[:MEASview<y>]:IRTime?`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<avg\_avg>,<avg\_min>,<avg\_max>

Where

<avg\_avg>::<NRf> is the average of the impulse response time.

<avg\_min>::<NRf> is the minimum of the impulse response time.

<avg\_max>::<NRf> is the maximum of impulse response time. Unit: seconds.

### Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:IRTime?` might return -2.35,-2.36,-2.34 for the impulse response time in the pulse statistics.

## **FETCH:PULSe:STATistics[:MEASview<y>]:MAX:PULSe:NUMber (Query Only)**

Returns the maximum value at a certain pulse number in the pulse statistics display.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

## Syntax

```
FETCH:PULSE:STATISTICS[:MEASview<y>]:MAX:PULSE:NUMBER?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<max_pulse_no>
```

Where

<max\_pulse\_no>::<NRf> is the maximum value at a pulse number.

## Examples

FETCH:PULSE:STATISTICS:MEASview2:MAX:PULSE:NUMBER? might return 15 for the maximum pulse number is 15.

## FETCh:PULSe:STATistics[:MEASview<y>]:MFReqerror (Query Only)

Returns the maximum frequency error in the pulse statistics. This command is valid when DISPLAY:PULSE:STATISTICS[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:MFReqerror?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<max_freq_err_avg>,<max_freq_err_min>,<max_freq_err_max>
```

Where

<max\_freq\_err\_avg>::<NRf> is the average of maximum frequency error.

<max\_freq\_err\_min>::<NRf> is the minimum of maximum frequency error.

<max\_freq\_err\_max>::<NRf> is the maximum of maximum frequency error. Unit: Hz.

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:MFRerror?` might return `645.0E+3,555.6E+3,738.8E+3` for the maximum frequency error in the pulse statistics.

## **FETCH:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMber (Query Only)**

Returns the minimum value at a certain pulse number in the pulse statistics.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number.

### Group

Fetch commands

### Syntax

```
FETCH:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMber?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<min\_pulse\_no>

Where

<min\_pulse\_no>::<NRf> is the minimum value at a pulse number.

### Examples

`FETCH:PULSE:STATISTICS:MIN:PULSE:NUMBER?` might return 1 for the minimum pulse number in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:MPHerror (Query Only)**

Returns the maximum phase error in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:MPHerror?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<max\_phase\_err\_avg>, <max\_phase\_err\_min>, <max\_phase\_err\_max>

Where

<max\_phase\_err\_avg>::<NRf> is the average of maximum phase error.

<max\_phase\_err\_min>::<NRf> is the minimum of maximum phase error.

<max\_phase\_err\_max>::<NRf> is the maximum of maximum phase error. Unit: Degrees.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:MPHerror? might return -11.47,-17.18, -7.61 for the maximum phase error in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:OVEDb (Query Only)

Returns the overshoot measurement result in dB in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd or TTRend.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:OVEDb?

## Related Commands

FETCh:PULSe:STATistics[:MEASview<y>]:OVERshoot?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<overshoot\_avg>, <overshoot\_min>, <overshoot\_max>

Where

<overshoot\_avg>::<NRf> is the average overshoot.

<overshoot\_min>::<NRf> is the minimum overshoot.

<overshoot\_max>::<NRf> is the maximum overshoot. Unit: dB.

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:OVEDb?` might return 0.3,0.1,0.5 for the overshoot measurement result in dB in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:OVERshoot (Query Only)**

Returns the overshoot in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND` or `TTRend`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:OVERshoot?`

### Related Commands

`FETCh:PULSe:STATistics[:MEASview<y>]:OVEDb?`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<overshoot_avg>,<overshoot_min>,<overshoot_max>`

Where

`<overshoot_avg>::<NRf>` is the average overshoot.

`<overshoot_min>::<NRf>` is the minimum overshoot.

`<overshoot_max>::<NRf>` is the maximum overshoot. Unit: Percent (%) by watts or volts.

### Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:OVERshoot?` might return 0.3,0.1,0.5 for the overshoot in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:PHDeviation (Query Only)**

Returns the phase deviation in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number



## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:PHDeviation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<phase_dev_avg>,<phase_dev_min>,<phase_dev_max>
```

Where

<phase\_dev\_avg>::<NRf> is the average phase deviation.

<phase\_dev\_min>::<NRf> is the minimum phase deviation.

<phase\_dev\_max>::<NRf> is the maximum phase deviation. Unit: Degrees.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:PHDeviation? might return 11.678,7.694, 17.374 for the phase deviation in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:PPFD (Query Only)

Returns the pulse-pulse frequency difference in the Pulse Statistics display.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:PPFD?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<NRf>
```

## Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:PPFD?` might return `365.4186562500E+3, -267.5911040000E+6, 247.1117760000E+6`, indicating the pulse-pulse frequency difference.

## **FETCh:PULSe:STATistics[:MEASview<y>]:PPFRequency (Query Only)**

Returns the pulse-pulse carrier frequency in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:PPFRequency?`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<pp_freq_avg>`, `<pp_freq_min>`, `<pp_freq_max>`

Where

`<pp_freq_avg>` :: `<NRf>` is the average pulse-pulse carrier frequency.

`<pp_freq_min>` :: `<NRf>` is the minimum pulse-pulse carrier frequency.

`<pp_freq_max>` :: `<NRf>` is the maximum pulse-pulse carrier frequency. Unit: Hz.

### Examples

`FETCH:PULSE:STATISTICS:MEASVIEW2:PPFRequency?` might return `1.135E+3,311.3E+3,-262.8E+3` for the pulse-pulse carrier frequency in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:PPOWer (Query Only)**

Returns the peak power in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:PPOWer?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<pk_power_avg>, <pk_power_min>, <pk_power_max>
```

Where

<pk\_power\_avg>: :<NRf> is the average peak power.

<pk\_power\_min>: :<NRf> is the minimum peak power.

<pk\_power\_max>: :<NRf> is the maximum peak power. Unit: dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:PULSe:STATistics:MEASview2:PPOWer? might return -2.273,-2.313,-2.235 for the peak power in the pulse statistics 2.

## FETCh:PULSe:STATistics[:MEASview<y>]:PPPD (Query Only)

Returns the overshoot measurement result in dB in the Pulse Statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd or TTRend.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:PPPD?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<NRf>
```

## Examples

`FETCh:PULSe:STATistics:MEASview2:PPPD?` might return `-1.1131216288, -151.8063964844, 130.5854492188`, indicating the pulse-pulse phase difference in dB.

## **FETCh:PULSe:STATistics[:MEASview<y>]:PPPHase (Query Only)**

Returns the pulse-pulse carrier phase in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND` or `TTRend`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:PPPHase?`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<pp_phase_avg>`, `<pp_phase_min>`, `<pp_phase_max>`

Where

`<pp_phase_avg>`: : `<NRf>` is the average pulse-pulse carrier phase.

`<pp_phase_min>`: : `<NRf>` is the minimum pulse-pulse carrier phase.

`<pp_phase_max>`: : `<NRf>` is the maximum pulse-pulse carrier phase. Unit: Degrees.

## Examples

`FETCh:PULSe:STATistics:MEASview2:PPPHase?` might return `-9.298E-3,-254.3E-3,311.7E-3` for the pulse-pulse carrier phase in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:RINTerval (Query Only)**

Returns the repetition interval in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:RINTerval?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<rep_int_avg>,<rep_int_min>,<rep_int_max>
```

Where

<rep\_int\_avg>::<NRf> is the average repetition interval.

<rep\_int\_min>::<NRf> is the minimum repetition interval.

<rep\_int\_max>::<NRf> is the maximum repetition interval. Unit: Seconds.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:RINTERVAL might return 240.5E-6,217.9E-6,281.2E-6 for the repetition interval in the pulse statistics 2.

## FETCh:PULSe:STATistics[:MEASview<y>]:RIPDb (Query Only)

Returns the ripple in dB in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:RIPDb?
```

## Related Commands

```
FETCh:PULSe:STATistics[:MEASview<y>]:RIPPlE?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<ripple_avg>,<ripple_min>,<ripple_max>
```

Where

<ripple\_avg>::<NRf> is the average ripple.

<ripple\_min>::<NRf> is the minimum ripple.

<ripple\_max>::<NRf> is the maximum ripple. Unit: dB.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:RIPDb? might return 160.4E-3,83.78E-3,287.7E-3 for the ripple in dB in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:RIPPlE (Query Only)

Returns the ripple in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:RIPPlE?

## Related Commands

FETCh:PULSe:STATistics[:MEASview<y>]:RIPDb?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<ripple\_avg>,<ripple\_min>,<ripple\_max>

Where

<ripple\_avg>::<NRf> is the average ripple.

<ripple\_min>::<NRf> is the minimum ripple.

<ripple\_max>::<NRf> is the maximum ripple. Unit: Percent (%) by watts or volts.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:RIPPlE? might return 160.4E-3,83.78E-3,287.7E-3 for the ripple in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:RISE (Query Only)**

Returns the rise time in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe:STATistics[:MEASview<y>]:RISE?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### **Returns**

```
<rise_avg>,<rise_min>,<rise_max>
```

Where

<rise\_avg>::<NRf> is the average rise time.

<rise\_min>::<NRf> is the minimum rise time.

<rise\_max>::<NRf> is the maximum rise time. Unit: Seconds.

### **Examples**

`FETCh:PULSe:STATISTICS:MEASVIEW2:RISE?` might return `105.4E-9,91.65E-9,116.2E-9` for the rise time in the pulse statistics.

## **FETCh:PULSe:STATistics[:MEASview<y>]:RMSFreqerror (Query Only)**

Returns the RMS frequency error in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number

### **Group**

Fetch commands

### **Syntax**

```
FETCh:PULSe:STATistics[:MEASview<y>]:RMSFreqerror?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<RMS\_freq\_err\_avg>, <RMS\_freq\_err\_min>, <RMS\_freq\_err\_max>

Where

<RMS\_freq\_err\_avg>::<NRf> is the average of RMS frequency error.

<RMS\_freq\_err\_min>::<NRf> is the minimum of RMS frequency error.

<RMS\_freq\_err\_max>::<NRf> is the maximum of RMS frequency error. Unit: Hz.

## Examples

FETCH:PULSE:STATISTICS:MEASVIEW2:RMSFreqerror? might return 63.67E+3, 45.49E+3, 81.28E+3 for the RMS frequency error in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:RMSPherror (Query Only)

Returns the RMS phase error in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:STATistics[:MEASview<y>]:RMSPherror?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<RMS\_phase\_err\_avg>, <RMS\_phase\_err\_min>, <RMS\_phase\_err\_max>

Where

<RMS\_phase\_err\_avg>::<NRf> is the average of RMS phase error.

<RMS\_phase\_err\_min>::<NRf> is the minimum of RMS phase error.

<RMS\_phase\_err\_max>::<NRf> is the maximum of RMS phase error. Unit: Degrees.



## Examples

`FETCh:PULSe:STATISTICS:MEASVIEW2:RMSPHERROR?` might return 1.032,604.5E-3, 1.606 for the RMS phase error in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:RRATe (Query Only)

Returns the repetition rate in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:RRATe?`

### Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<rep_rate_avg>`, `<rep_rate_min>`, `<rep_rate_max>`

Where

`<rep_rate_avg>::<NRf>` is the average repetition rate.

`<rep_rate_min>::<NRf>` is the minimum repetition rate.

`<rep_rate_max>::<NRf>` is the maximum repetition rate. Unit: Hz.

### Examples

`FETCh:PULSe:STATistics[:MEASview<y>]:RRATe?` might return 62.50E+3,62.49E+3, 62.52E+3 for the repetition rate in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:WIDTH (Query Only)

Returns the pulse width in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:WIDTh?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<width_avg>,<width_min>,<width_max>
```

Where

<width\_avg>::<Nrf> is the average pulse width.

<width\_min>::<Nrf> is the minimum pulse width.

<width\_max>::<Nrf> is the maximum pulse width. Unit: Seconds.

## Examples

FETCh:PULSe:STATistics[:MEASview<y>]:WIDTh? might return 4.636E-6,4.630E-6, 4.643E-6 for the pulse width in the pulse statistics.

## FETCh:PULSe:STATistics[:MEASview<y>]:X? (Query Only)

Returns the horizontal values of the trace data of the Pulse Statistics measurement selected by the DISPLAY:PULSe:SElect:RESult command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT command before executing this query.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:STATistics[:MEASview<y>]:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
#<num_digit><num_byte><data1><data2>...data<n>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the horizontal value of the `STATistics:MEASview<y>` graph at the nth data point.

The unit is trace number (Plot = Trend), Hz (Plot = FFT), second (Plot = Time Trend) or current measurement unit (Plot = Histogram).

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:PULSe:STATistics:MEASview2:X?` might return `#264xxxx`. (64-byte data) for the `STATistics:MEASview2` trace of the pulse width measurement when `DISPlay:PULSe:SElect:RESult` is set to `WIDTH`.

## FETCh:PULSe:STATistics[:MEASview<y>]:XY? (Query Only)

Returns the horizontal and vertical values of the trace data of the pulse statistics measurement selected by the `DISPlay:PULSe:SElect:RESult` command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` command before executing this query.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

`FETCh:PULSe:STATistics[:MEASview<y>]:XY?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`#<num_digit><num_byte><data1><data2>...data<n>`

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the horizontal value of the `STATistics:MEASview<y>` graph at the nth data point.

<data (n+1)> is the vertical value of the `STATistics:MEASview<y>` graph at the nth data point.

For horizontal values, the unit is trace number (Plot = Trend), Hz (Plot = FFT), second (Plot = Time Trend) or current measurement unit (Plot = Histogram).

For vertical values, the unit is current measurement unit (Plot = Trend or Time Trend), dB (Plot = FFT), count (Plot = Histogram)

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:PULSe:STATistics:MEASview1:XY?` might return `#264xxxx...` (64-byte data) for the `STATistics:MEASview1` trace of the pulse width measurement when `DISPlay:PULSe:SElect:RESult` is set to `WIDTh`.

## FETCh:PULSe:STATistics[:MEASview<y>][:Y]? (Query Only)

Returns the trace data of the pulse statistics measurement selected by the `DISPlay:PULSe:SElect:RESult` command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` command before executing this query.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Fetch commands

## Syntax

`FETCh:PULSe:STATistics[:MEASview<y>][:Y]?`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the amplitude at the `n`th data point.

The unit is current measurement unit (Plot = Trend or Time Trend), dB (Plot = FFT), count or percent (Plot = Histogram)

4-byte little endian floating-point format specified in IEEE 488.2.

The unit of power is selected by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCh:PULSe:STATISTICS:MEASVIEW2:Y?` might return `#264xxxx...` (64-byte data) for the `STATistics 2` trace of the pulse width measurement when `DISPlay:PULSe:SElect:RESult` is set to `WIDTh`.

## FETCh:PULSe:TRACe[:MEASview<y>]:X

Returns the time values of the pulse trace. Use the `DISPlay:PULSe:SElect:NUMBer` command to select the pulse, and the `DISPlay:PULSe:SElect:RESult` command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:TRACe[:MEASview<y>]:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
#<num_digit><num_byte><X(1)><X(2)>...<X(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<X(n)> is the time in seconds at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:PULSe:TRACe:MEASVIEW1:X?` might return `#43204xxxx...` (3204-byte data) for the time values of the trace in pulse trace measurement view 1.

## FETCh:PULSe:TRACe[:MEASview<y>]:XY

Returns the horizontal (time) and vertical value pairs of the pulse trace. Use the `DISPlay:PULSe:SElect:NUMBer` command to select the pulse, and the `DISPlay:PULSe:SElect:RESult` command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Fetch commands

## Syntax

```
FETCh:PULSe:TRACe[:MEASview<y>]:XY?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

#<num\_digit><num\_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the horizontal value (time in seconds) and vertical value pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

The vertical unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The vertical unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

FETCH:PULSE:TRACE:MEASVIEW1:XY? might return #43204xxxx... (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace measurement view 1.

## FETCh:PULSe:TRACe[:MEASview<y>]:Y

Returns the vertical values of the pulse trace. Use the DISPLAY:PULSe:SElect:NUMBER command to select the pulse, and the DISPLAY:PULSe:SElect:RESult command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Fetch commands

## Syntax

FETCh:PULSe:TRACe[:MEASview<y>]:Y?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

#<num\_digit><num\_byte><y(1)><y(2)>...<y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the amplitude (dBm) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCH:PULSE:TRACE:MEASVIEW1:Y?` might return `#43204xxxx...` (3204-byte data) for the vertical values of the pulse trace measurement view 1.

## FETCh:RFIN:IQ? (Query Only)

Returns time-domain IQ data for a specific acquisition data record. You can set a range of IQ pairs optionally. The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

## Conditions

Measurement views: All

## Group

Fetch commands

## Syntax

```
FETCh:RFIN:IQ? <rec_ID>[,<begin_num>,<end_num>]
```

## Arguments

`<rec_ID>` specifies the acquisition data record ID number.

`<begin_num>` specifies the beginning number of IQ pairs.

`<end_num>` specifies the end number of IQ pairs.

Use the `FETCh:RFIN:RECOrd:IDS?` query to get the beginning and end ID's of acquisition data records.

## Returns

```
#<num_digit><num_byte><I(1)><Q(1)><I(2)><Q(2)>...<I(n)><Q(n)>
```

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<I(n)>` and `<Q(n)>` are the time-domain IQ data pair.

4-byte little endian floating-point format specified in IEEE 488.2.

To find out the range of IQ pairs, use the `FETCh:RFIN:IQ:HEADer?` query. The returned value of `<num_sample>` is the number of IQ pairs.



**Note:** Do not fetch a large number of IQ pairs because of memory limitation.

## Examples

`FETCH:RFIN:IQ? 10,25,350` might return `#43848xxxx...` (3848-byte data) for the record #10, ranging from 25th to 350th IQ pair.

## FETCh:RFIN:IQ:HEADer? (Query Only)

Returns the header information for a specific acquisition data record. The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

### Conditions

Measurement views: All

### Group

Fetch commands

### Syntax

```
FETCh:RFIN:IQ:HEADer? <rec_ID>
```

### Arguments

<rec\_ID> specifies the acquisition data record ID number.

Use the FETCh:RFIN:RECOrd:IDS? query to get the beginning and end ID's of acquisition data records.

### Returns

```
<span>,<sampling_freq>,<num_sample>,<center_freq>,<acq_BW>,  
<ref_level>,<trig_pos>,<time_stamp>,<FastFrame_ID>
```

Where

<span>::<NR3> is the span in Hz.

<sampling\_freq>::<NR3> is the sampling frequency in Hz.

<num\_sample>::<NR3> is the number of samples.

<center\_freq>::<NR3> is the center frequency in Hz.

<acq\_BW>::<NR3> is the acquisition bandwidth in Hz.

<ref\_level>::<NR3> is the reference level in dBm.

<trig\_pos>::<NR3> is the trigger position in seconds.

<time\_stamp>::<string> is the time stamp.

<FastFrame\_ID>::<NR1> is the Fast Frame ID: Zero or positive number. If the Fast Frame is disabled, the ID will be negative.

### Examples

```
FETCh:RFIN:IQ:HEADer? 10 might return 40.0E+6,50.0E+6,4.027E+3, 1.5E+9,40.0E+6,0.0,20.242E-6,"10/31/2007 11:18:32 AM",-1, indicating Span: 40 MHz, Sampling frequency: 50 MHz, Number of samples: 4027, Center frequency: 1.5 GHz, Acquisition bandwidth: 40 MHz, Reference level: 0 dBm, Trigger position: 20.242  $\mu$ s, Time stamp: 10/31/2007 11:18:32 AM, and Fast Frame ID: -1
```

## FETCh:RFIN:IQ:SCALE? (Query Only)

Returns the internal RF linear data scaling factor contained in the .tiq file header. The scaling factor can be used to convert digital IQ output (Option 05) values into real IQ values.



## Conditions

Measurement views: All

## Group

Fetch commands

## Syntax

```
FETCh:RFIN:IQ:SCALE?
```

## Related Commands

```
FETCh:RFIN:RECOrd:IDS?
```

## Arguments

None

## Returns

<NR3> The RF linear data scaling factor.

## Examples

FETCh:RFIN:IQ:SCALE? might return 19.553E-6 for the scaling factor.

## FETCh:RFIN:IQ:VHEader? (Query Only)

Returns the verbose header information for a specific acquisition data record. The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

## Conditions

Measurement views: All

## Group

Fetch commands

## Syntax

```
FETCh:RFIN:IQ:VHEader? <rec_ID>
```

## Arguments

<rec\_ID> specifies the acquisition data record ID number.

Use the FETCh:RFIN:RECOrd:IDS? query to get the beginning and end ID's of acquisition data records.

## Returns

```
<span>,<sampling_freq>,<num_sample>,<center_freq>,<acq_BW>,<ref_level>,<trig_pos>,<time_stamp>,<FastFrame_ID>
```

Where

<span>::<NR3> is the span in Hz.

<sampling\_freq>::<NR3> is the sampling frequency in Hz.

<num\_sample>::<NR3> is the number of samples.

<center\_freq>::<NR3> is the center frequency in Hz.

<acq\_BW>::<NR3> is the acquisition bandwidth in Hz.

<ref\_level>::<NR3> is the reference level in dBm.

<trig\_pos>::<NR3> is the trigger position in seconds.

<time\_stamp>::<string> is the time stamp.

<FastFrame\_ID>::<NR1> is the Fast Frame ID: Zero or positive number. If the Fast Frame is disabled, the ID will be negative.

## Examples

FETCH:RFIN:IQ:VHEADER? 10 might return 110.0000000000E+6,150.0000000000E+6,4.0270000000E+3,2.7000000000E+9,110.0000000000E+6,0.0000,20.2420000000E-6, "7/9/2013 8:38:25.9305101 -07:00:00",-1, indicating Span: 110 MHz, Sampling frequency: 150 MHz, Number of samples: 4027, Center frequency: 2.7 GHz, Acquisition bandwidth: 110 MHz, Reference level: 0 dBm, Trigger position: 20.242  $\mu$ s, Time stamp: 10/31/2007 11:18:32 AM, Fast Frame ID: -1

## FETCH:RFIN:RECORD:IDS? (Query Only)

Returns the beginning and end ID numbers of acquisition data.



**Note:** The instrument needs to be in stopped mode. If not in stopped mode, it returns the execution error (-200).

## Conditions

Measurement views: All

## Group

Fetch commands

## Syntax

FETCH:RFIN:RECORD:IDS?

## Arguments

None

## Returns

<begin\_ID>, <end\_ID>

Where

<begin\_ID>::<NR1> is the beginning ID of acquisition data.

<end\_ID>::<NR1> is the end ID of acquisition data.



**Note:** "-1,-1" is returned when the span changes and the acquisition is armed, but the acquisition has not yet occurred.

## Examples

`FETCH:RFIN:RECORD:IDS?` might return 1,147, indicating the beginning and end ID's of acquisition data are 1 and 147, respectively.

## FETCh:SEM:MHITs:COUNT? (Query Only)

Returns the number of rows in the results table in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:MHITs:COUNT?`

### Arguments

None

### Returns

`<NRf>` The number of rows in the table.

## Examples

`FETCH:SEM:MHITS:COUNT?` might return 7 indicating there are seven rows in the results table.

## FETCh:SEM:MHITs<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:MHITs<x>:AMPLitude:ABSolute?`

### Arguments

None

## Returns

<NRf> Absolute amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`FETCh:SEM:MHITs2:AMPLitude:ABSolute?` might return `-73.17` indicating that the absolute amplitude of signal in row 2 is `-73.17` dBm.

## FETCh:SEM:MHITs<x>:AMPLitude:RELative? (Query Only)

Returns the relative amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

`FETCh:SEM:MHITs<x>:AMPLitude:RELative?`

## Arguments

None

## Returns

<NRf> Relative amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`FETCh:SEM:MHITs2:AMPLitude:RELative?` might return `-91.81` indicating that the relative amplitude of mask hit signal in row 2 is `-91.81` dBm.

## FETCh:SEM:MHITs<x>:FREQuency? (Query Only)

Returns the frequency of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

```
FETCH:SEM:MHITs<x>:FREQuency?
```

## Arguments

None

## Returns

<NRf> Relative amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

FETCH:SEM:MHITs4:FREQuency? might return 2.235E+6 indicating the frequency for mask hit signal in row 4 is 2.235 MHz.

## FETCH:SEM:MHITs<x>:INTEg:ABSolute? (Query Only)

Returns the absolute integration amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

```
FETCH:SEM:MHITs<x>:INTEg:ABSolute?
```

## Arguments

None

## Returns

<NRf> Absolute integration amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

FETCH:SEM:MHITs4:INTEg:ABSolute? might return -75.14 indicating that the absolute integration amplitude for mask hit signal in row 4 is -75.14 dBm.

## FETCH:SEM:MHITs<x>:INTEg:RELative? (Query Only)

Returns the relative integration amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

`FETCH:SEM:MHITs<x>:INTEg:RELative?`

## Arguments

None

## Returns

<NRf> Relative integration amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`FETCH:SEM:MHITs4:INTEg:RELative?` might return `-85.14` indicating that the relative integration amplitude for mask hit signal in row 4 is `-85.14` dBm.

## **FETCH:SEM:MHITs<x>:MARGin:ABSolute? (Query Only)**

Returns the absolute margin for the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

`FETCH:SEM:MHITs<x>:MARGin:ABSolute?`

## Arguments

None

## Returns

<NRf> Absolute margin of the specified signal in dB.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`FETCH:SEM:MHITs4:MARGin:ABSolute?` might return `-62.33` indicating that the absolute margin for mask hit signal in row 4 is `-62.33` dB.

## **FETCh:SEM:MHITs<x>:MARGin:RELative? (Query Only)**

Returns the relative margin for the specified mask hit signal in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Fetch commands

### **Syntax**

```
FETCh:SEM:MHITs<x>:MARGin:RELative?
```

### **Arguments**

None

### **Returns**

<NRf> Relative margin of the specified signal in dB.

<x> specifies the row in the results table, where the range is from 1 to 12.

### **Examples**

FETCh:SEM:MHITs4:MARGin:RELative? might return -62.33 indicating that the relative margin for mask hit signal in row 4 is -62.33 dB.

## **FETCh:SEM:MHITs<x>:OFFSet? (Query Only)**

Returns the offset for the specified mask hit signal in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Fetch commands

### **Syntax**

```
FETCh:SEM:MHITs<x>:OFFSet?
```

### **Arguments**

None

### **Returns**

<NRf> specifies the mask hit offset.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`FETCh:SEM:MHITs4:OFFSet?` might return BL indicating the offset for mask hit signal in row 4 is BL.

## FETCh:SEM:PASS? (Query Only)

Returns the pass/fail limit test result in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:PASS?`

### Arguments

None

### Returns

0 (fail) or 1 (pass)

## Examples

`FETCh:SEM:PASS?` might return 1, indicating that the limit test was successful.

## FETCh:SEM:REF:POWer? (Query Only)

Returns the reference power level in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:REF:POWer?`

### Arguments

None

### Returns

`<NRf>` specifies the reference power.



## Examples

`FETCH:SEM:REF:POWER?` might return `-16.92 dBm` indicating that the reference power level is `-16.92 dBm`.

## FETCh:SEM:SPECTrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:SPECTrum:X?`

### Arguments

None

### Returns

`#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<x(n)>` is the frequency (Hz) at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

`FETCH:SEM:SPECTRUM:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of the spectrum trace.

## FETCh:SEM:SPECTrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Fetch commands

### Syntax

`FETCh:SEM:SPECTrum:XY?`

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the frequency (Hz) and amplitude (dBm) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:SEM:SPECTRUM:XY? might return #516020xxxx... (16020-byte data) for the frequency and amplitude pairs of the spectrum trace.

## FETCh:SEM:SPECTrum[:Y]? (Query Only)

Returns the amplitude of the spectrum trace in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Fetch commands

## Syntax

```
FETCh:SEM:SPECTrum[:Y]?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the frequency (Hz) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:SEM:SPECTRUM[:Y]? might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace.

## FETCh:SGRam[:MEASview<y>] (Query Only)

Returns trace data of a line in the spectrogram. The line is selected using the `TRACe:SGRam[:MEASview<y>]:SElect:LINE` command.

### Conditions

Measurement views: Spectrogram

### Group

Fetch commands

### Syntax

```
FETCh:SGRam[:MEASview<y>]?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

```
#<num_digit><num_byte><data (1)><data (2)>...<data (n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude in dBm at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the `[SENSe]:POWer:UNITs` command.

### Examples

`FETCh:SGRam:MEASview1?` might return `#43204xxxx...` (3204-byte data) for the line in the spectrogram.

## FETCh:SIGNalvu:ACQuisition:COUPling? (Query Only)

Returns the coupling value of the acquisition.

### Conditions

Should be connected to MSO Query

### Group

Fetch commands

### Syntax

```
FETCh:SIGNalvu:ACQuisition:COUPling
```

## Related Commands

FETCh:SIGNalvu:ACQuisition:COUPling?

## Arguments

None

## Returns

<max>::<NRf> is the maximum phase in degrees.

## Examples

FETCh:SIGNalvu:ACQuisition:COUPling? might return 153.8, indicating the maximum phase is 153.8 °.

## FETCh:SIGNalvu:ACQuisition:MAX:RT:BW? (Query Only)

Queries the maximum real time bandwidth value of the instrument.

## Conditons

Should be connected to MSO Query

## Group

Fetch commands

## Syntax

FETCh:SIGNalvu:ACQuisition:MAX:RT:BW

## Related Commands

FETCh:SIGNalvu:ACQuisition:MAX:RT:BW?

## Arguments

None

## Returns

<max>::<NRf> is the maximum phase in degrees.

## Examples

FETCh:SIGNalvu:ACQuisition:MAX:RT:BW? might return 153.8, indicating the maximum phase is 153.8 °.

## **FETCH:SIGNalvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:HIGHSide:MAXFreq? (Query only)**

Returns the maximum range of input center frequency in case of high injection side. The parameter <x>=1 to 8 represents the range of the source.

### **Conditions**

Should be connected to MSO/LPD/RSA/SX to query.

Source keyword is optional, if not present Source1 will be default source.

### **Group**

Fetch commands

### **Syntax**

```
FETCH:SIGNalvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:HIGHSide:MAXFreq?
```

### **Related Commands**

```
:FETCH:SIGNalvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:HIGHSide:MINFreq?
```

### **Arguments**

None

### **Returns**

<max>::<NRF> returns the maximum value of the centre frequency range in case of high injection side.

### **Examples**

```
:FETCH:SIGNalvu:ACQuisition:SOURce1:DOWNconverter:INJEction:HIGHSide:MAXFreq? Might  
return 9.5000000000E+9, indicating the maximum range of the input center frequency in case of high injection side.
```

## **FETCH:SIGNalvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:HIGHSide:MINFreq? (Query only)**

Returns the minimum range of the input center frequency in case of high injection side. The parameter <x>=1 to 8 represents the range of the source.

### **Conditions**

Should be connected to MSO/LPD/RSA/SX to query.

Source keyword is optional, if not present, Source1 is default source.

### **Group**

Fetch commands

### **Syntax**

```
FETCH:SIGNalvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:HIGHSide:MINFreq?
```

## Arguments

None

## Returns

<max> : : <Nrf> returns the minimum value of the centre frequency range in case of high injection side.

## Examples

:FETCH:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:INJEction:HIGHSide:MINFreq? Might return 2.0000000000E+9, indicating the minimum range of the input center frequency in case of high injection side.

## **FETCH:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:LOWSide:MAXFreq? (Query only)**

Returns the maximum range of the input center frequency in case of low injection side. The parameter <x>=1 to 8 represents the range of the source.

## Conditions

Should be connected to MSO/LPD/RSA/SX to query.

Source keyword is optional, if not present, Source1 will be default source.

## Group

Fetch commands

## Syntax

FETCH:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:LOWSide:MAXFreq?

## Related Commands

FETCH:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:LOWSide:MINFreq?

## Arguments

None

## Returns

<max> : : <Nrf> returns the maximum value of centre frequency range in case of low injection side.

## Examples

:FETCH:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:INJEction:LOWSide:MAXFreq? Might return 9.5000000000E+9, indicating the maximum range of the input center frequency in case of low injection side.

## **FETCH:SIGNAlvu:ACQuisition[:SOURce<x>]: DOWNconverter:INJEction:LOWSide:MINFreq? (Query only)**

Returns the minimum range of input center frequency in case of low injection side. The parameter <x>=1 to 8 represents the range of the source.

### **Conditions**

Should be connected to MSO/LPD/RSA/SX to query.

Source keyword is optional, if not present, Source1 will be default source

### **Group**

Fetch commands

### **Syntax**

```
FETCH:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:INJEction:LOWSide:MINFreq?
```

### **Arguments**

None

### **Returns**

<max> : : <NRF> returns the minimum value of the centre frequency range in case of low injection side.

### **Examples**

```
:FETCH:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:INJEction:LOWSide:MINFreq? Might  
return 2.0000000000E+9, indicating the minimum range of the input center frequency in case of low injection side.
```

## **FETCh:SIGNAlvu:ACQuisition:TERMination? (Query Only)**

Queries the termination value of the acquisition.

### **Conditons**

Should be connected to MSO Query

### **Group**

Fetch commands

### **Syntax**

```
FETCh:SIGNAlvu:ACQuisition:TERMination
```

### **Related Commands**

```
FETCh:SIGNAlvu:ACQuisition:TERMination?
```

### **Arguments**

None

## Returns

<max> : : <Nrf> is the maximum phase in degrees.

## Examples

FETCh:SIGNAlvu:ACQuisition:TERMination? might return 153.8, indicating the maximum phase is 153.8 °.

## FETCh:SPECTrum[:MEASview<y>]:TRACe<x>? (Query Only)

Returns the trace data in the Spectrum measurement.

The parameter <x> = 1 to 5.



**Note:** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

## Conditions

Measurement views: Spectrum

## Group

Fetch commands

## Syntax

FETCh:SPECTrum[:MEASview<y>]:TRACe<x>?

## Related Commands

TRACe<x>:SPECTrum

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude in dBm at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:SPECTRUM:MEASview1:TRACE1? might return #43204xxxx... (3204-byte data) for Trace 1 in the Spectrum measurement.



## FETCh:SPURious:CARRier:POWer? (Query Only)

Returns the carrier power in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

```
FETCh:SPURious:CARRier:POWer?
```

### Arguments

None

### Returns

<NRf> Carrier power in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

### Examples

FETCh:SPURIOUS:CARRIER:POWER? might return 4.227, indicating that the carrier power is 4.227 dBm.

## FETCh:SPURious:COUNT? (Query Only)

Returns the number of spurious signals in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

```
FETCh:SPURious:COUNT?
```

### Arguments

None

### Returns

<NRf> The number of spurious signals.

## Examples

`FETCH:SPURIOUS:COUNT?` might return 4, indicating that the spurious count is 4.

## FETCh:SPURious:PASS? (Query Only)

Returns the pass/fail limit test result in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

`FETCh:SPURious:PASS?`

### Arguments

None

### Returns

0 (fail) or 1 (pass).

## Examples

`FETCH:SPURIOUS:PASS?` might return 1, indicating that the limit test was successful.

## FETCh:SPURious:SPECTrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

`FETCh:SPURious:SPECTrum:X?`

### Arguments

None

### Returns

`#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x (n) > is the frequency (Hz) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:SPURIOUS:SPECTRUM:X? might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

## FETCh:SPURious:SPECTrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

FETCh:SPURious:SPECTrum:XY?

### Arguments

None

### Returns

#<num\_digit><num\_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x (n) ><y (n) > is the frequency (Hz) and amplitude (dBm) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

The amplitude unit can be changed by the [SENSe]:POWER:UNITs command.

## Examples

FETCH:SPURIOUS:SPECTRUM:XY? might return #516020xxxx... (16020-byte data) for the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

## FETCh:SPURious:SPECTrum[:Y]? (Query Only)

Returns the amplitudes of the spectrum trace in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

```
FETCh:SPURious:SPECTrum[:Y]?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><y(1)><y(2)>...<y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the amplitude (dBm) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

FETCh:SPURIOUS:SPECTRUM:Y? might return #516020xxxx... (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

## FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

```
FETCh:SPURious:SPUR<x>:AMPLitude:ABSolute?
```

## Arguments

None

## Returns

<NRf> Absolute amplitude of the specified spurious signal in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

`FETCH:SPURIOUS:SPUR1:AMPLITUDE:ABSOLUTE?` might return `-19.782`, indicating that the absolute amplitude of Spurious #1 is `-19.782` dBm.

## **FETCh:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)**

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

`FETCh:SPURious:SPUR<x>:AMPLitude:RELative?`

### Arguments

None

### Returns

`<NRf>` Relative amplitude of the specified spurious signal in dB. Use the `[SENSe]:SPURious:REFErence` command to set the power reference.

## Examples

`FETCH:SPURIOUS:SPUR1:AMPLITUDE:RELATIVE?` might return `-9.782`, indicating that the relative amplitude of Spurious #1 is `-9.782` dB.

## **FETCh:SPURious:SPUR<x>:FREQuency:ABSolute? (Query Only)**

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Fetch commands

### Syntax

`FETCh:SPURious:SPUR<x>:FREQuency:ABSolute?`

### Arguments

None

## Returns

<NRf> Absolute frequency of the spurious signal in Hz.

## Examples

`FETCH:SPURIOUS:SPUR1:FREQUENCY:ABSOLUTE?` might return `2.235E+9`, indicating that the absolute frequency of Spurious #1 is 2.235 GHz.

## FETCh:SPURious:SPUR<x>:FREQuency:RELative? (Query Only)

Returns the relative frequency of the specified spurious signal to the carrier in the Spurious measurement. This command is valid when `[SENSE]:SPURious:REFerence` is set to `CARRier`.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

`FETCh:SPURious:SPUR<x>:FREQuency:RELative?`

## Arguments

None

## Returns

<NRf> Relative frequency of the spurious signal to the carrier in Hz.

## Examples

`FETCH:SPURIOUS:SPUR1:FREQUENCY:RELATIVE` might return `3.634E+6`, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

## FETCh:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

`FETCh:SPURious:SPUR<x>:LIMit:ABSolute?`

## Arguments

None

## Returns

<NRf> Absolute amplitude of the limit for the specified spurious signal in dBm.

The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

FETCH:SPURIOUS:SPUR1:LIMIT:ABSOLUTE? might return -50.0, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

## FETCh:SPURious:SPUR<x>:LIMit:RELative? (Query Only)

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

```
FETCh:SPURious:SPUR<x>:LIMit:RELative?
```

## Arguments

None

## Returns

<NRf> Relative amplitude of the limit for the specified spurious signal in dB. Use the [SENSE]:SPURious:REFerence command to set the power reference.

## Examples

FETCH:SPURIOUS:SPUR1:LIMIT:RELATIVE? might return -10.0, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

## FETCh:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)

Returns whether the specified spurious signal exceeds the limit or not.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

`FETCh:SPURious:SPUR<x>:LIMit:VIOLation?`

## Arguments

None

## Returns

0 (under the limit) or 1 (over the limit).

## Examples

`FETCh:SPURIOUS:SPUR1:LIMIT:VIOLATION?` might return 1, indicating that Spurious #1 exceeds the limit.

## FETCh:SPURious:SPUR<x>:RANGe? (Query Only)

Returns the frequency range in which the specified spurious signal occurred.

## Conditions

Measurement views: Spurious

## Group

Fetch commands

## Syntax

`FETCh:SPURious:SPUR<x>:RANGe?`

## Arguments

None

## Returns

`<string>` "A" to "T" representing Range A to T, respectively.

## Examples

`FETCh:SPURIOUS:SPUR1:RANGE` might return "E", indicating that Spurious #1 is in Range E.

## FETCh:SQUality[:MEASview<y>]:FREQuency:DEViation? (Query only)

Returns the frequency deviation in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality



MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:FREQuency:DEVIation?
```

## Arguments

None

## Returns

<NRf> Frequency deviation in Hz.

## Examples

FETCh:SQUality:MEASview1:FREQuency:DEVIation? might return 12.68E+3, indicating the frequency deviation is 12.68 kHz for measurement view1.

## FETCh:SQUality[:MEASview<y>]:FREQuency:DEVIation:TABLE? (Query only)

Returns the number of columns and the values in the frequency deviation table for a signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:FREQuency:DEVIation:TABLE?
```

## Arguments

None

## Returns

<Dev\_Num>,{<Freq\_dev\_Max>,<Freq\_dev\_Min>,<Freq\_dev\_Avg>}

Where

<Dev\_Num>::<NR1> is the number of columns in the frequency deviation table.

= 2 (2FSK), 4 (4FSK, C4FM), 8 (8FSK), or 16 (16FSK)

<Freq\_dev\_Max>::<NRf> is the maximum frequency deviation in Hz.

<Freq\_dev\_Min>::<NRf> is the minimum frequency deviation in Hz.

<Freq\_dev\_Avg>::<NRf> is the average frequency deviation in Hz.

The dataset <Freq\_dev\_Max>, <Freq\_dev\_Min>, <Freq\_dev\_Avg> is returned for each symbol in ascending order of its level (for example, in order of symbol -3, -1, +1, and +3 for 4FSK).

## Examples

FETCh:SQUality:MEASview1FREQuency:DEVIation:TABLE? might return

2,1.257E+3,1.039E+3,1.162E+3,1.586E+3,1.298E+3,1.425E+3 for the frequency signal, populating the results table as follows.

Deviations	-1	+1
Maximum	1.257 kHz	1.586 kHz
Minimum	1.039 kHz	1.298 kHz
Average	1.162 kHz	1.425 kHz

## FETCh:SQUality[:MEASview<y>]:FREQuency:ERRor? (Query only)

Returns the frequency error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### Group

Fetch commands

### Syntax

FETCh:SQUality[:MEASview<y>]:FREQuency:ERRor?

### Arguments

None

### Returns

<NRf> Frequency error in Hz.

## Examples

`FETCh:SQQuality:MEASview1:FREQuency:ERRor?` might return `612.043E+3`, indicating that the frequency error is 612.043 kHz for measurement view 1.

## **FETCh:SQQuality[:MEASview<y>]:GAIN:IMBalance? (Query only)**

Returns the gain imbalance in the signal quality measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQQuality[:MEASview<y>]:GAIN:IMBalance?
```

## Arguments

None

## Returns

`<NRf>` Gain imbalance in dB.

## Examples

`FETCh:SQQuality[:MEASview1]:GAIN:IMBalance?` might return `-57.746E-3`, indicating that the gain imbalance is -0.057746 dB for measurement view 1.

## **FETCh:SQQuality[:MEASview<y>]:ORIGin:OFFSet? (Query only)**

Returns the origin offset in the signal quality measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:ORIGin:OFFSet?`

## Arguments

None

## Returns

<NRf> Origin offset in dB.

## Examples

`FETCh:SQUality:MEASview1:ORIGin:OFFSet?` might return -44.968, indicating that the origin offset is -44.968 dB for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:EVM? (Query only)**

Returns the peak EVM (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:PEAK:EVM?`

## Arguments

None

## Returns

<NRf> Peak EVM in percent (%).

## Examples

`FETCh:SQUality:MEASview1:PEAK:EVM?` might return 4.276, indicating that the peak EVM is 4.276% for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB? (Query only)**

Returns the peak EVM (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB?
```

## Arguments

None

## Returns

<NRf> Peak EVM in dB.

## Examples

FETCh:SQUality:MEASview1:PEAK:EVM:DB? might return -27.358, indicating that the peak EVM is -27.358 dB for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet? (Query only)

Returns the peak offset EVM (dB) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet?
```

## Arguments

None

## Returns

<NRf> Peak offset EVM in dB.

## Examples

FETCh:SQUality:MEASview1:PEAK:EVM:DB:OFFSet? might return -37.624, indicating the peak offset EVM is -37.624 dB for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation? (Query only)

Returns the time at which the EVM is peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation?

## Arguments

None

## Returns

<NRf> The time in symbol number at which the EVM is peak.

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

FETCh:SQUality:MEASview1:PEAK:EVM:LOCation? might return 68.000, indicating that the EVM is peak at symbol #68.000 for measurement view1.

## FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet? (Query only)

Returns the time at which the offset EVM is peak.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet?
```

## Arguments

None

## Returns

<NRf> The time in symbol number at which the offset EVM is peak.

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

FETCh:SQUality:MEASview1:PEAK:EVM:LOCation:OFFSet? might return 123.00, indicating that the offset EVM is peak at symbol #123 for measurement view1.

## FETCh:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet? (Query only)

Returns the peak offset EVM (%) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet?
```

## Arguments

None

## Returns

<NRf> Peak offset EVM in percent (%).

## Examples

`FETCh:SQUality:MEASview1:PEAK:EVM:OFFSet?` might return 1.298, indicating the peak offset EVM is 1.298% for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:FERRor? (Query only)**

Returns the peak FSK error in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to FSK2, FSK4, FSK8, or FSK16.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:PEAK:FERRor?`

## Arguments

None

## Returns

<NRf> Peak FSK error in percent (%).

## Examples

`FETCh:SQUality:MEASview1:PEAK:FERRor?` might return 9.136, indicating the peak FSK error is 9.136% for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:DB? (Query only)**

Returns the peak magnitude error (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality



MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:DB?
```

## Arguments

None

## Returns

<NRf> Peak magnitude error in dB.

## Examples

FETCh:SQUality:MEASview1:PEAK:EVM:DB? might return -27.358, indicating that the peak EVM is -27.358 dB for measurement view1.

## FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:LOCation? (Query only)

Returns the time at which the magnitude error is peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:PEAK:MERRor:LOCation?
```

## Arguments

None

## Returns

<NRf> The time in symbol number at which the magnitude error is peak.

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

`FETCh:SQUality:MEASview1:PEAK:MERRor:LOCation?` might return 68.000, indicating that the magnitude error is peak at symbol #68 for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:MERRor? (Query only)**

Returns the peak magnitude error (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:PEAK:MERRor?`

## Arguments

None

## Returns

<NRf> Peak magnitude error in percent (%).

## Examples

`FETCh:SQUality:MEASview1:PEAK:MERRor?` might return 3.595, indicating that the peak magnitude error is 3.595% for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:PEAK:PERRor:LOCation? (Query only)**

Returns the time at which the phase error is peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCH:SQUALity[:MEASview<y>]:PEAK:PERRor:LOCation?
```

## Arguments

None

## Returns

<NRf> The time in symbol number at which the phase error is peak.

The unit can be changed by the [SENSE]:DDEMod:TIME:UNITs command.

## Examples

FETCH:SQUALity:MEASview1:PEAK:PERRor:LOCation? might return 68.000, indicating that the phase error is peak at symbol #68 for measurement view1.

## FETCH:SQUALity[:MEASview<y>]:PEAK:PERRor? (Query only)

Returns the peak phase error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCH:SQUALity[:MEASview<y>]:PEAK:PERRor?
```

## Arguments

None

## Returns

<NRf> Peak phase error in degrees.

## Examples

FETCH:SQUALity:MEASview1:PEAK:PERRor? might return 1.907, indicating that the peak phase error is 1.907 ° for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:QUADrature:ERRor? (Query only)**

Returns the quadrature error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### **Group**

Fetch commands

### **Syntax**

```
FETCh:SQUality[:MEASview<y>]:QUADrature:ERRor?
```

### **Arguments**

None

### **Returns**

<NRf> Quadrature error in degrees.

### **Examples**

FETCh:SQUality:MEASview1:QUADrature:ERRor? might return -14.264E-3, indicating that the quadrature error is -0.014264° for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:RHO? (Query only)**

Returns the  $\rho$  (waveform quality) value in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### **Group**

Fetch commands

### **Syntax**

```
FETCh:SQUality[:MEASview<y>]:RHO?
```

## Arguments

None

## Returns

<NRf>  $\rho$  value.

## Examples

`FETCh:SQUality:MEASview1:RHO?` might return `998.703E-3`, indicating that  $\rho$  is 0.998703 for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:RMS:EVM? (Query only)**

Returns the RMS EVM (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:EVM?
```

## Arguments

None

## Returns

<NRf> RMS EVM in percent (%).

## Examples

`FETCh:SQUality:MEASview1:RMS:EVM?` might return `2.417`, indicating that the RMS EVM is 2.417% for measurement view 1.

## **FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB? (Query only)**

Returns the RMS EVM (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB.

## Examples

FETCh:SQUality:MEASview1:RMS:EVM:DB? might return -32.356, indicating that the RMS EVM is -32.356 dB for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB:OFFSet? (Query only)

Returns the RMS offset EVM (dB) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:EVM:DB:OFFSet?
```

## Arguments

None

## Returns

<NRf> RMS offset EVM in dB.

## Examples

`FETCh:SQUality:MEASview1:RMS:EVM:DB:OFFSet?` might return `-41.276`, indicating the RMS offset EVM is `-41.276` dB for measurement view 1.

## **FETCh:SQUality[:MEASview<y>]:RMS:EVM:OFFSet? (Query only)**

Returns the RMS offset EVM (%) in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to `OQPSK` or `SOQPSK`.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:RMS:EVM:OFFSet?`

## Arguments

None

## Returns

<NRf> RMS offset EVM in percent (%).

## Examples

`FETCh:SQUality:MEASview1:RMS:EVM:OFFSet?` might return `0.783`, indicating the RMS offset EVM is `0.783%` for measurement view 1.

## **FETCh:SQUality[:MEASview<y>]:RMS:FERRor? (Query only)**

Returns the RMS FSK error percent in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to `FSK2`, `FSK4`, `FSK8`, or `FSK16`.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:FERRor?
```

## Arguments

None

## Returns

<NRf> RMS FSK error in percent (%).

## Examples

FETCh:SQUality:MEASview1:RMS:FERRor? might return 1.815, indicating the RMS FSK error is 1.815% for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:RMS:MER:DB? (Query only)

Returns the RMS MER (Modulation Error Ratio) in dB in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:MER:DB?
```

## Arguments

None

## Returns

<NRf> RMS MER in dB.



## Examples

`FETCh:SQUality:MEASview1:RMS:MER:DB?` might return 27.394, indicating that the RMS MER is 27.394 dB for measurement view1.

## **FETCh:SQUality[:MEASview<y>]:RMS:MERRor? (Query only)**

Returns the RMS magnitude error (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:RMS:MERRor?
```

## Arguments

None

## Returns

<NRf> RMS magnitude error in percent (%).

## Examples

`FETCh:SQUality:MEASview1:RMS:MERRor?` might return 1.837, indicating that the RMS magnitude error is 1.837% for measurement view 1.

## **FETCh:SQUality[:MEASview<y>]:RMS:MERRor:DB? (Query only)**

Returns the RMS magnitude error (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:RMS:MERRor:DB?`

## Arguments

None

## Returns

<NRf> RMS MERRor in dB.

## Examples

`FETCh:SQUality:MEASview1:RMS:MERRor:DB?` might return -34.706, indicating that the magnitude error is -34.706 dB for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:RMS:PERRor? (Query only)

Returns the RMS phase error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:RMS:PERRor?`

## Arguments

None

## Returns

<NRf> RMS phase error in degrees.

## Examples

`FETCh:SQUality:MEASview1:RMS:PERRor?` might return 893.472E-3, indicating that the RMS phase error is 0.893472 °.

## FETCh:SQUality[:MEASview<y>]:SYMBOL:LENGth? (Query only)

Returns the number of analyzed symbols.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:SYMBOL:LENGth?
```

## Returns

<NR1> Indicates the length of the synch word in symbols.

## Examples

FETCh:SQUality:MEASview1:SYMBOL:LENGth? might return 3, indicating the length is three symbols for measurement view 1.

## FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE? (Query only)

Returns the value of the calculated symbol rate in Hz.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

It is valid when the modulation type is 2|4|8|16FSK and [ :SENSe ] :DDEMod :SYMBOL :RATE :SEARCh is ON.

## Group

Fetch commands

## Syntax

```
FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SYMBOL:RATE:SEARCh](#)

## Returns

<NRf> is the calculated symbol rate in Hz.

## Examples

`FETCh:SQUality:MEASview1:SYMBOL:RATE?` might return `95.24E+3`, indicating the calculated symbol rate is 95.24 kHz for measurement view 1.

## **FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE:ERRor? (Query only)**

Returns the value of the symbol rate error in percent (%).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

It is valid when the modulation type is 2|4|8|16FSK and `[ :SENSe ] :DDEMod:SYMBOL:RATE:SEARCh` is ON.

## Group

Fetch commands

## Syntax

`FETCh:SQUality[:MEASview<y>]:SYMBOL:RATE:ERRor?`

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SYMBOL:RATE:SEARCh](#)

## Returns

<NRf> is the symbol error percent (%).

## Examples

`FETCh:SQUality:MEASview1:SYMBOL:RATE:ERRor?` might return `-0.002`, indicating that the symbol rate error is -0.002% for measurement view 1.

## **FETCh:TDIagram[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the Trellis diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Fetch commands

## Syntax

```
FETCh:TDIagram[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

FETCh:TDIagram:MEASview2:FERRor? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

## FETCh:TDIagram[:MEASview<y>]:TRACe? (Query only)

Returns the Trellis diagram trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Fetch commands

## Syntax

```
FETCh:TDIagram[:MEASview<y>]:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> is the phase in degrees and the time in (symbols) pair at the n<sup>th</sup> data point. The 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:TDIagram:MEASview2:FDEVIATION?` might return `#3160xxxx...` (160-byte data) for the Trellis diagram trace.

## FETCh:TOVerview[:MEASview<y>] (Query Only)

Returns the trace data in the time overview.

### Conditions

Measurement views: Time overview with specified Measurement view number

### Group

Fetch commands

### Syntax

`FETCh:TOVerview[:MEASview<y>]?`

### Arguments

The parameter `<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the amplitude in dBm at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2. The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`FETCh:TOVERVIEW:MEASVIEW1?` might return `#43204xxxx...` (3204-byte data) for the trace in the time overview.

## FETCh:TXGain:AVERage? (Query Only)

Returns the mean value of the trace.

### Conditions

Measurement view: Transmission Gain

### Group

Fetch commands

### Syntax

`FETCh:TXGain:AVERage?`

## Returns

<NRf> is the averaged value of the trace points in dB.

## Examples

`FETCH:TXGAIN:AVERAGE?` might return 2.3, indicating that the mean value of the trace is 2.3 dB.

## FETCh:TXGain:MAX? (Query Only)

Returns the maximum trace value.

## Conditions

Measurement view: Transmission Gain

## Group

Fetch commands

## Syntax

`FETCh:TXGain:MAX?`

## Returns

<NRf> is the maximum value of the trace in dB.

## Examples

`FETCH:TXGAIN:MAX?` might return -3.45, indicating the maximum trace value is -3.45 dB.

## FETCh:TXGain:MAXLocation? (Query Only)

Returns the frequency of the maximum trace value.

## Conditions

Measurement view: Transmission Gain

## Group

Fetch commands

## Syntax

`FETCh:TXGain:MAXLocation?`

## Returns

<NRf> is the frequency at the maximum trace value.

## Examples

`FETCH:TXGAIN:MAXLOCATION?` might return 1E9, indicating the maximum trace value is at 1 GHz.

## **FETCh:TXGain:MIN? (Query Only)**

Returns the minimum trace value.

### **Conditions**

Measurement view: Transmission Gain

### **Group**

Fetch commands

### **Syntax**

FETCh:TXGain:MIN?

### **Returns**

<NRf> is the frequency at the minimum trace value in dB.

### **Examples**

FETCh:TXGAIN:MIN? might return -30, indicating the minimum trace value is -30 dB.

## **FETCh:TXGain:MINLocation? (Query Only)**

Returns the frequency of the minimum trace value.

### **Conditions**

Measurement view: Transmission Gain

### **Group**

Fetch commands

### **Syntax**

FETCh:TXGain:MINLocation?

### **Returns**

<NRf> is the frequency at the minimum trace value.

### **Examples**

FETCh:TXGAIN:MINLOCATION? might return 1E6, indicating the minimum trace value is at 1 MHz.

## **FETCh:TXGain:TRACe<x>? (Query Only)**

Returns the trace data of the selected trace.

### **Conditions**

Measurement view: Transmission Gain Where <x> is the trace 1, 2, or 3.



## Group

Fetch commands

## Syntax

```
FETCh:TXGain:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude in dB at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCh:TXGain:TRACe1? might return #43204xxxx... (3204 byte data) for Trace 1 in the measurement.

## FETCh:WLAN:AD:CONSte:DATA:EVM? (Query Only)

Queries the maximum EVM value in data Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AD:CONSte:DATA:EVM?
```

## Arguments

None

## Returns

<NRf> = EVM value

## Examples

FETCh:WLAN:AD:CONSTE:DATA:EVM? might return -42.482, indicating the EVM value present in the analysis data is -42.482 dB.

## FETCh:WLAN:AD:CONSte:DATA:EVM:PEAK? (Query Only)

Queries the EVM Peak value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:CONStE:DATA:EVM:PEAK?`

## Arguments

None

## Returns

<NRf> = EVM peak value

## Examples

`FETCH:WLAN:AD:CONSTE:DATA:EVM:PEAK?` might return `-35.325`, indicating the EVM peak value is `-35.325` dB.

## **FETCh:WLAN:AD:CONStE:DATA:EVM:PEAK:SYMBol? (Query Only)**

Queries the EVM peak symbol value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:CONStE:DATA:EVM:PEAK:SYMBol?`

## Arguments

None

## Returns

<NRf> = symbol value

## Examples

`FETCH:WLAN:AD:CONSTE:DATA:EVM:PEAK:SYMBOL?` might return `6.312`, indicating the EVM peak symbol value is `6.312`.

## **FETCh:WLAN:AD:CONStE:FERRor? (Query Only)**

Queries the frequency Error value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:CONStE:FERRor?`

## Arguments

None

## Returns

<NR3>= frequency error

## Examples

`FETCH:WLAN:AD:CONSTE:FERROR?` might return `3.344+3`, indicating the frequency error value is 3.344 kHz.

## FETCh:WLAN:AD:CONStE:IQDC:OFFSet? (Query Only)

Queries IQ Origin Offset value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:CONStE:IQDC:OFFSet?`

## Arguments

None

## Returns

<NRf> = IQ Origin Offset value

## Examples

`FETCH:WLAN:AD:CONSTE:IQDC:OFFSET?` might return `-51.841`, indicating the IQ Origin Offset value is -51.841 dB.

## FETCh:WLAN:AD:EVM:TRACe? (Query Only)

Returns the EVM versus time trace data.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Fetch commands

## Syntax

FETCH:WLAN:AD:EVM:TRACe?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data(n)> is the EVM value at the nth data point in percent (%),

<data(n)> follows 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:WLAN:AD:EVM:TRACE? might return #42036xxxx... (2036-byte data), indicating the EVM versus time trace data.

## FETCH:WLAN:AD:SUMMARY:AVERAGE:BURSt:COUNT (Query Only)

Queries the current burst number out of the total burst in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

FETCH:WLAN:AD:SUMMARY:AVERAGE:BURSt:COUNT?

## Arguments

None

## Returns

<NR1>

## Examples

`FETCH:WLAN:AD:SUMMARY:AVERAGE:BURST:COUNT` might return 1, indicating the current burst count is 1 out of the total burst.

## **FETCh:WLAN:AD:SUMMArY:AVERAge:BURSt:TOTAL:COUNT (Query Only)**

Queries the total burst count to be averaged in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMArY:AVERAge:BURSt:TOTAL:COUNT?`

### Arguments

None

### Returns

<NR1>

## Examples

`FETCH:WLAN:AD:SUMMARY:AVERAGE:BURST:TOTAL:COUNT` might return 1, indicating the total burst count to be averaged is 1.

## **FETCh:WLAN:AD:SUMMArY:ESNR (Query Only)**

Returns the value of Estimated SNR value in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMArY:ESNR ?`

### Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:ESNR` might return 36.856, indicating the estimated SNR value is 36.856 dB.

## **FETCh:WLAN:AD:SUMMArY:EVM:DATA:AVERAge (Query Only)**

Queries the Avg EVM (Data) value in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:DATA:AVERAge?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE` might return -40.7035275551, indicating the Avg EVM (Data) value is -40.70.

## **FETCh:WLAN:AD:SUMMArY:EVM:DATA:AVERAge:STATUs (Query Only)**

Queries the Avg EVM (Data) status in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:DATA:AVERAge:STATUs?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE:STATUS` might return PASS, indicating the Avg EVM test as passed.

## **FETCh:WLAN:AD:SUMMArY:EVM:DATA:MAX (Query Only)**

Queries the maximum EVM (Data) value in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:DATA:MAX?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:DATA:MAX` might return 922.1968241036E-3, indicating the maximum EVM (Data) value is 922.1968241036E-3.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:CEF:EVM (Query Only)**

Queries the EVM value in the CEF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:CEF:EVM?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:CEF:EVM` might return 6.9680400193, indicating the EVM value in the CEF region of the packet is 6.96 dB.

## **FETCh:WLAN:AD:SUMMary:EVM:PACKet:CEF:PEAK (Query Only)**

Queries the peak error value at the given symbol in CEF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:CEF:PEAK?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:CEF:PEAK` might return 13.7763872743, indicating the peak error at the given symbol in CEF region of the packet is 13.77 dB.

## **FETCh:WLAN:AD:SUMMary:EVM:PACKet:CEF:SYMBol (Query Only)**

Queries the symbol at which the peak error occurred in the CEF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary



## Group

Fetch commands

## Syntax

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:CEF:SYMBOL?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:CEF:SYMBOL` might return `7.402000000E+3`, indicating the peak error occurred in symbol 7.40 in the CEF region of the packet.

## **FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM (Query Only)**

Queries the EVM value in the DATA region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM` might return `-33.5668742100`, indicating the EVM value in the DATA region of the packet is -33.56 dB.

## **FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:PEAK (Query Only)**

Queries the peak error value at the given symbol in the DATA region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:PEAK?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:PEAK` might return `-26.5964736883`, indicating the peak error value is `-26.59` dB at the given symbol in DATA region of the packet.

## **FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:STATus (Query Only)**

Queries the status of the DATA packet in Summary display in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable.

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:STATUS` might return `PASS`, indicating the `DATA` packet test as passed.

## **FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:SYMBol (Query Only)**

Queries the symbol at which the peak error occurred in the `DATA` region of the packet in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:DATA:SYMBol?`

### Arguments

None

### Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:DATA:SYMBOL` might return `4.661000000E+3`, indicating that the symbol peak error occurred at 4.66 in the `DATA` region of the packet.

## **FETCh:WLAN:AD:SUMMary:EVM:PACKet:GUARd:EVM (Query Only)**

Queries the EVM value in Guard region of the Packet in the 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMary:EVM:PACKet:GUARd:EVM ?`

### Arguments

None

### Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:EVM` might return `-65.4983500315`, indicating the EVM value in the Guard region of the packet is `-65.49` dB.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:GUARd:PEAK (Query Only)**

Queries the peak error value at the given symbol in the Guard region of the packet in the 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:PACKet:GUARd:PEAK?`

### Arguments

None

### Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:PEAK` might return `-26.5964736883`, indicating the peak error at the given symbol in the Guard region of the packet is `-26.59` dB.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:GUARd:SYMBol (Query Only)**

Queries the symbol at which the peak error occurred in the Guard region of the packet in the 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:PACKet:GUARd:SYMBol?`

### Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:SYMBOL` might return `3.334000000E+3`, indicating the peak error occurred at symbol 3.33 in the Guard region of the packet.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:EVM (Query Only)**

Queries the EVM value in the Header region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:EVM?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:EVM` might return `-67.1981081169`, indicating the EVM value in the Header region of the packet is `-67.19` dB.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:PEAK (Query Only)**

Queries the peak error at the given symbol in the Header region if the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:PEAK?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:PEAK` might return `-60.9603086224`, indicating the peak error at the given symbol is `-60.96` dB in the Header region of the packet.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:SYMBol (Query Only)**

Queries the symbol at which the peak Error occurred in the Header region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:EVM:PACKet:HEADer:SYMBol?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:SYMBOL` might return `3.3970000000E+3`, indicating the symbol peak error occurred is `3.397` in Header region of the packet.

## **FETCh:WLAN:AD:SUMMArY:EVM:PACKet:STF:EVM (Query Only)**

Queries the EVM value in the STF region of the packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AD:SUMMary:EVM:PACKet:STF:EVM?
```

## Arguments

None

## Returns

<NRf>

## Examples

FETCH:WLAN:AD:SUMMARY:EVM:PACKET:STF:EVM might return -22.9729890175, indicating the EVM value in the STF region of the packet is -22.97 dB.

## FETCh:WLAN:AD:SUMMary:EVM:PACKet:STF:PEAK (Query Only)

Queries the peak error value at the given symbol in the STF region of the packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AD:SUMMary:EVM:PACKet:STF:PEAK?
```

## Arguments

None

## Returns

<NRf>

## Examples

FETCH:WLAN:AD:SUMMARY:EVM:PACKET:STF:PEAK might return 38.8123661280, indicating the peak error at the given symbol in the STF region of the packet is 38.81 dB.

## FETCh:WLAN:AD:SUMMary:EVM:PACKet:STF:SYMBol (Query Only)

Queries the symbol at which the peak error occurred in the STF region of the packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:STF:SYMBOL?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:EVM:PACKET:STF:SYMBOL` might return 0.0, indicating the peak error occurred at the 0.0 symbol in the STF region of the packet.

## **FETCH:WLAN:AD:SUMMARY:FERROR:AVERAGE (Query Only)**

Queries the Avg Frequency Error value in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCH:WLAN:AD:SUMMARY:FERROR:AVERAGE?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:FERROR:AVERAGE` might return `-114.1332879317E-3`, indicating the Avg Frequency error is `-114.1332879317E-3` ppm.

## **FETCH:WLAN:AD:SUMMARY:FERROR:AVERAGE:STATUS (Query Only)**

Queries the Avg Frequency error status in 802.11ad Summary display.



## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AD:SUMMary:FERRor:AVERage:STATus?
```

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

## Examples

FETCh:WLAN:AD:SUMMary:FERRor:AVERage:STATus might return FAIL, indicating the Avg Frequency error test failed.

## FETCh:WLAN:AD:SUMMary:FERRor:HERTZ (Query Only)

Queries the Frequency Error value in 802.11ad Summary display in Hz.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AD:SUMMary:FERRor:HERTZ?
```

## Arguments

None

## Returns

<NRf>

## Examples

FETCh:WLAN:AD:SUMMary:FERRor:HERTZ might return 2.875E+3 , indicating the Frequency Error value is 2.875E+3 Hz.

## **FETCh:WLAN:AD:SUMMary:FERRor:MAX (Query Only)**

Queries the Max Frequency error value in the 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AD:SUMMary:FERRor:MAX?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

FETCh:WLAN:AD:SUMMary:FERRor:MAX might return -114.1332879317E-3, indicating the max frequency error is -114.1332879317E-3 ppm.

## **FETCh:WLAN:AD:SUMMary:FERRor:PPM (Query Only)**

Queries the Frequency Error value in 802.11ad Summary display in ppm.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AD:SUMMary:FERRor:PPM?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

FETCh:WLAN:AD:SUMMary:FERRor:PPM might return 575.023E-3, indicating the Frequency Error value is 575.023E-3 ppm.

## **FETCh:WLAN:AD:SUMMary:FERRor:STATus (Query Only)**

Queries the Frequency error status in the 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AD:SUMMary:FERRor:STATus?

### **Arguments**

None

### **Returns**

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

### **Examples**

FETCh:WLAN:AD:SUMMary:FERRor:STATus might return PASS, indicating the frequency error test passed.

## **FETCh:WLAN:AD:SUMMary:IQ:GAIN:IMBALance (Query Only)**

Queries IQ Gain Imbalance value in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AD:SUMMary:IQ:GAIN:IMBALance?

### **Arguments**

None

### **Returns**

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:IQ:GAIN:IMBALANCE` might return `69.2700818181E-3`, indicating IQ Gain Imbalance value is `69.2700818181E-3 %`.

## **FETCh:WLAN:AD:SUMMary:IQ:ORIGin:OFFSet (Query Only)**

Queries IQ Origin Offset value in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMary:IQ:ORIGin:OFFSet?`

### Arguments

None

### Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:IQ:ORIGIN:OFFSET` might return `-160.0000000000`, indicating the IQ Origin Offset value is `-160.0 dB`.

## **FETCh:WLAN:AD:SUMMary:IQ:QUADrature:ERROr (Query Only)**

Queries IQ Quadrature Error value in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AD:SUMMary:IQ:QUADrature:ERROr?`

### Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:IQ:QUADRATURE:ERROR` might return `-220.5967903137E-3`, indicating IQ Quadrature Error value is `-220.5967903137E-3` degrees.

## FETCh:WLAN:AD:SUMMary:PACKet:AGGRegation (Query Only)

Queries the value of Aggregation packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:PACKet:AGGRegation?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:AGGREGATION` might return `"0"`, indicating the Aggregation packet value is 0.

## FETCh:WLAN:AD:SUMMary:PACKet:BTR (Query Only)

Queries the value of Beam Tracking Request packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMary:PACKet:BTR?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:BTR` might return 0, indicating the Beam Tracking Request packet value is 0.

## **FETCh:WLAN:AD:SUMMArY:PACKet:HCS (Query Only)**

Queries the value of HCS packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:HCS?`

## Arguments

None

## Returns

<String>

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:HCS` might return "1001010000001000", indicating the HCS packet value is 1001010000001000.

## **FETCh:WLAN:AD:SUMMArY:PACKet:LENGth (Query Only)**

Queries the value of Length packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:LENGth?`

## Arguments

None

## Returns

<String>

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:LENGTH` might return "1000", indicating the Length packet value is 1000.

## FETCh:WLAN:AD:SUMMArY:PACKet:MCS (Query Only)

Queries the value of MCS packet information in Summary display in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:MCS?`

## Arguments

None

## Returns

<NR1> Range is 0 to 12

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:MCS` might return 5, indicating the MCS packet value is 5.

## FETCh:WLAN:AD:SUMMArY:PACKet:PPDU (Query Only)

Queries the value of Additional PPDU packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:PPDU?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:PPDU` might return "0", indicating the Additional PDU packet value is 0.

## **FETCh:WLAN:AD:SUMMArY:PACKet:RSSI (Query Only)**

Queries the value of Last RSSI packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:RSSI?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:RSSI` might return "0000", indicating the Last RSSI packet value is 0000.

## **FETCh:WLAN:AD:SUMMArY:PACKet:SCRAMbler:INITIALIZATION (Query Only)**

Queries the value of Scrambler initialization packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:SCRAMbler:INITIALIZATION?`

## Arguments

None



## Returns

<String>

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION` might return "0101110", indicating the Scrambler Initialization packet value is 0101110.

## FETCh:WLAN:AD:SUMMArY:PACKet:TAROUND (Query Only)

Queries the value of the Turnaround packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:TAROUND?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:TAROUND` might return "1", indicating the Turnaround packet value is 1.

## FETCh:WLAN:AD:SUMMArY:PACKet:TLEN (Query Only)

Queries the value of Training Length packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:TLEN?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:TLEN` might return 21, indicating the Training Length packet value is 21.

## **FETCh:WLAN:AD:SUMMArY:PACKet:TYPE (Query Only)**

Queries the value of Packet Type information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:PACKet:TYPE?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`FETCH:WLAN:AD:SUMMARY:PACKET:TYPE` might return "0", indicating the Packet Type value is 0.

## **FETCh:WLAN:AD:SUMMArY:RCPI (Query Only)**

Queries the RCPI value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:RCPI?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:RCPI` might return 207.0000000000, indicating the RCPI is 207.

## **FETCh:WLAN:AD:SUMMArY:RF:POWer (Query Only)**

Queries the RF Output Power value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AD:SUMMArY:RF:POWer?`

## Arguments

None

## Returns

<NRf>

## Examples

`FETCH:WLAN:AD:SUMMARY:RF:POWER` might return -6.3434391022, indicating the RF Output Power is -6.3434391022 dBm.

## **FETCh:WLAN:AY:CONStE:DATA:EVM:PEAK? (Query Only)**

Query the Data Peak Error value in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AY:CONStE:DATA:EVM:PEAK?`

## Returns

<NRf>

You must select the units in the EVM tab.

## Examples

FETC:WLAN:AY:CONS:DATA:EVM:PEAK? may return -35.325, which means -35.325 is the Data Peak Error value.

## FETCh:WLAN:AY:CONSte:DATA:EVM:PEAK:SYMBol? (Query Only)

Query the Symbol position that contains the maximum Data EVM value in the Constellation display.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:CONSte:DATA:EVM:PEAK:SYMBol?

### Returns

<NRf>

## Examples

FETC:WLAN:AY:CONS:DATA:EVM:PEAK:SYMB? may return 6.312e+0003, which indicates that the 6312 Symbol contains the maximum Data EVM value.

## FETCh:WLAN:AY:CONSte:DATA:EVM? (Query Only)

Query the maximum EVM value present in the analysis data in the Constellation display.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:CONSte:DATA:EVM?

### Returns

<NRf>

You must select the units in the EVM tab.

## Examples

FETC:WLAN:AY:CONS:DATA:EVM? may return -42.482, which means the maximum EVM value in the analysis data is -42.482.

## FETCh:WLAN:AY:CONStE:FERRor? (Query Only)

Query the Frequency Error value in the Constellation display.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:CONStE:FERRor?
```

### Returns

<NRf>

Units: Hz

## Examples

FETC:WLAN:AY:CONS:FERR? may return 0.0000, which indicates that the Frequency Error value is 0.0000 Hz.

## FETCh:WLAN:AY:CONStE:FREQuency:ERROr? (Query Only)

Query the Frequency Error value in the Constellation display.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:CONStE:FREQuency:ERROr?
```

### Returns

<NRf>

Units: Hz

## Examples

FETC:WLAN:AY:CONS:FREQ:ERRO? may return 0.0000, which means the Frequency Error value is 0.0000 Hz.

## **FETCh:WLAN:AY:CONStE:IQDC:OFFSet? (Query Only)**

Query the IQ Origin Offset value in the Constellation display.

### **Conditions**

Measurement view: 802.11ay Analysis Constellation

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:CONStE:IQDC:OFFSet?

### **Returns**

<NRf>

Units: dB

### **Examples**

FETCh:WLAN:AY:CONStE:IQDC:OFFSet? may return -51.841, which indicates that the IQ Origin Offset value is -51.841 dB.

## **FETCh:WLAN:AY:EVM:TRACe? (Query Only)**

Query the EVM versus Time trace data after doing a single run.

### **Conditions**

Measurement views: 802.11ay Analysis EVM

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:EVM:TRACe?

### **Returns**

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the EVM value at the  $n^{\text{th}}$  data point in percentage (%).

<data (n)> follows 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

FETC:WLAN:AY:EVM:TRAC? may return #42036xxx...(2036-byte data) for the EVM versus time trace.

## FETCh:WLAN:AY:SUMMary:AVERage:BURSt:COUNT? (Query Only)

Query the current Burst Count in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:AVERage:BURSt:COUNT?

### Returns

<NR1>

## Examples

FETC:WLAN:AY:SUMM:AVER:BURS:COUN? may return 1, which indicates that the current Burst Count is 1.

## FETCh:WLAN:AY:SUMMary:AVERage:BURSt:TOTAL:COUNT? (Query Only)

Query the Total Burst Count to be averaged.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:AVERage:BURSt:TOTAL:COUNT?

### Returns

<NR1>

## Examples

FETC:WLAN:AY:SUMM:AVER:BURS:TOTA:COUN? may return 1, which indicates that the Total Burst Count to be averaged is 1.

## **FETCh:WLAN:AY:SUMMary:ESNR? (Query Only)**

Query the Estimated SNR (Signal-to-Noise Ratio) value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:ESNR?

### **Returns**

<NRf>

Units: dB

### **Examples**

FETC:WLAN:AY:SUMM:ESNR? may return 34.50, which indicates that the Estimated SNR value is 34.50 dB.

## **FETCh:WLAN:AY:SUMMary:EVM:DATA:AVERage? (Query Only)**

Query the Average EVM (Data) value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:EVM:DATA:AVERage?

### **Returns**

<NRf>

Units: dB or % (based on the selected EVM unit)

### **Examples**

FETC:WLAN:AY:SUMM:EVM:DATA:AVER? may return -40.7035275551, which indicates that the Average EVM (Data) value is -40.7035275551%.

## **FETCh:WLAN:AY:SUMMary:EVM:DATA:AVERage:STATus? (Query Only)**

Query the Average EVM (Data) Status in the Summary display.



## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:DATA:AVERage:STATus?
```

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:EVM:DATA:AVER:STAT? may return PASS, FAIL, or NA.

## FETCh:WLAN:AY:SUMMary:EVM:DATA:MAX? (Query Only)

Query the Maximum EVM (Data) value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:DATA:MAX?
```

## Returns

<NRf>

Units: dB or % (based on the selected EVM unit)

## Examples

FETC:WLAN:AY:SUMM:EVM:DATA:MAX? may return 922.1968241036e-3, which indicates that the Maximum EVM (Data) value is 922.1968241036e-3 dB.

## FETCh:WLAN:AY:SUMMary:EVM:DATA:SIGma? (Query Only)

Query the Standard Deviation (1-Sigma) of the EVM Data.

When the EVM unit is in dB, it queries the Standard Deviation after converting evm values to the dB scale.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:DATA:SIGma?

## Returns

<NRf>

Units: dB

## Examples

FETC:WLAN:AY:SUMM:EVM:DATA:SIG? may return 1.8950, which indicates that the 1-Sigma EVM Data value is 1.8950 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:EVM? (Query Only)

Query the EVM value that occurs in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:EVM?

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:DATA:EVM? may return -33.5668742100, which indicates that the EVM value is -33.5668742100%.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:PEAK? (Query Only)

Query the value of the Peak Error that occurs at the given Symbol in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:PEAK?

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:DATA:PEAK? may return -26.5964736883, which indicates that the value of the Peak Error is -26.5964736883 at the given Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:STATus? (Query Only)

Query the status of the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:STATus?

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:DATA:STAT? may return PASS, FAIL, or NA.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:SYMBol? (Query Only)

Query the Symbol at which the Peak Error occurs in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:DATA:SYMBol?

## Returns

<NRf>

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:DATA:SYMB? may return 4.661000000e+3, which indicates that the Peak Error occurs at the 4.661000000e+3 Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:EVM? (Query Only)

Query the EVM value in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:EVM?

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:EVM? may return 6.9680400193, which indicates that the EVM value is -6.9680400193 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:PEAK? (Query Only)

Query the Peak Error that occurs at the given Symbol in the 11ay 2.16 GHz BW and EDMG CEF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:PEAK?

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:PEAK? may return 13.7763872743, which indicates that the Peak Error value is 13.7763872743% at the given Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:SYMBol? (Query Only)

Query the Symbol at which the Peak Error occurs in the 11ay 2.16 GHz BW and EDMG CEF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:SYMBol?

## Returns

<NRf>

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:SYMB? may return 7.402000000e+3, which indicates that the Peak Error occurs at the 7.402000000e+3 Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:EVM? (Query Only)

Query the EVM value in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:EVM?

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:EVM?` may return `-22.9729890175`, which indicates that the EVM value is `-22.9729890175` dB.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:PEAK? (Query Only)**

Query the Peak Error that occurs at the given Symbol in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:PEAK?`

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:PEAK?` may return `38.8123661280`, which indicates that the Peak Error value is `38.8123661280%` at the given Symbol.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:SYMBol? (Query Only)**

Query the Symbol at which the Peak Error occurs in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:SYMBol?`

### Returns

<NRf>

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:SYMB? may return 0.0000, which indicates that the Peak Error occurs at the 0.0000 Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:EVM? (Query Only)

Query the EVM value in the Guard Region of the packet in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:EVM?
```

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:GUAR:EVM? may return -65.4983500315, which indicates that the value of the EVM is -65.4983500315 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:PEAK? (Query Only)

Query the Peak Error that occurs at the given Symbol in the Guard Region of the packet in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:PEAK?
```

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:GUAR:PEAK? may return -26.5964736883, which indicates that the value of the Peak Error is -26.5964736883 at the given Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:SYMBol? (Query Only)

Query the Symbol at which the Peak Error occurs in the Guard Region of the packet in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:GUARd:SYMBol?

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:GUAR:SYMB? may return 3.334000000e+3, which indicates that the Peak Error occurs at the 3.334000000e+3 Symbol.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:EVM? (Query Only)

Query the EVM value in the Header Region of the packet (excluding GI symbols) in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:EVM?

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)



## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:HEAD:EVM?` may return -67.1981081169, which indicates that the EVM value in the Header Region is -67.198108116 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:PEAK? (Query Only)

Query the Peak Error that occurs at the given Symbol in Header Region (excluding GI symbols) of the packet in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:PEAK?`

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:HEAD:PEAK?` may return -60.9603086224, which indicates that the Peak Error value is -60.9603086224 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:SYMBol? (Query Only)

Query the Symbol at which the Peak Error occurs in the Header Region of the packet (excluding GI symbols) in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:HEADer:SYMBol?`

### Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:HEAD:SYMB?` may return `3.397000000e+3`, which indicates that the peak error occurs at the `3.397000000e+3` Symbol.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: A1Header:EVM? (Query**

Query the value of the Pre-EDGM1 Header-A EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: A1Header:EVM?`

### Returns

`<string>`

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:PRE1E:A1H:EVM?` may return `-34.3702`, which indicates that the value of the Pre-EDGM1 Header-A EVM packet is `-34.3702` dB.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: CEF1:EVM? (Query Only)**

Query the value of the Pre-EDMG1 CEF EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: CEF1:EVM?`

### Returns

`<string>`

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE1E:CEF1:EVM? may return -34.5666, which indicates that the value of the Pre-EDMG1 CEF EVM packet is -34.5666 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: GUARd1:EVM? (Query

Query the value of the Pre-EDMG1 EVM Guard region of the packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: GUARd1:EVM?
```

### Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE1ELGUAR1:EVM? may return -34.1327, which indicates that the value of the Pre-EDMG1 EVM Guard Region is -34.1327 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg :HEADer1:EVM? (Query

Query the value of the Pre-EDMG1 L-Header EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

```
FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg :HEADer1:EVM?
```

### Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE1E:HEAD1:EVM? may return -34.4326, which indicates that the value of the Pre-EDMG1 L-Header EVM packet is -34.4326 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: STF1:EVM? (Query Only)

Query the value of the Pre-EDMG1 STF EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg: STF1:EVM?

### Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE1E:STF1:EVM? may return -33.6325, which indicates that the value of the Pre-EDMG1 STF1 EVM packet is -33.6325 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: A2Header:EVM? (Query

Query the value of the Pre-EDMG2 EDMG Header-A EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: A2Header:EVM?

### Returns

<string>

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:PRE2E:A2H:EVM?` may return `-34.37`, which indicates that the value of the Pre-EDMG2 EDMG-Header-A EVM packet is `-34.37` dB.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: CEF2:EVM? (Query Only)**

Query the value of the Pre-EDMG2 CEF EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: CEF2:EVM?`

### Returns

`<string>`

Units: dB or % (based on the EVM unit selected)

## Examples

`FETC:WLAN:AY:SUMM:EVM:PACK:PRE2E:CEF2:EVM?` may return `-34.5665`, which indicates that the value of the Pre-EDMG2 CEF EVM packet is `-34.5665` dB.

## **FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: GUARd2:EVM? (Query**

Query the value of the Pre-EDMG2 Guard EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

`FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: GUARd2:EVM?`

### Returns

`<string>`

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE2E:GUAR:EVM? may return -33.137, which indicates that the value of the Pre-EDMG2 Guard EVM packet is -33.137 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: HEADer2:EVM? (Query

Query the value of the Pre-EDMG2 L-Header EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: HEADer2:EVM?

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE2E:HEAD2:EVM? may return -34.4321, which indicates that the value of the Pre-EDMG2 L-Header EVM packet is -34.4321 dB.

## FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: STF2:EVM? (Query Only)

Query the value of the Pre-EDMG2 STF EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg: STF2:EVM?

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

FETC:WLAN:AY:SUMM:EVM:PACK:PRE2E:STF2:EVM? may return -33.6324, which indicates that the value of the Pre-EDMG2 STF2 EVM packet is -33.6324 dB.

## FETCh:WLAN:AY:SUMMary:FERRor:AVERage? (Query Only)

Query the Average Frequency Error value in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:FERRor:AVERage?

### Returns

<NRf>

Units: ppm

## Examples

FETC:WLAN:AY:SUMM:FERR:AVER? may return -114.1332879317e-3, which indicates that the Average Frequency Error value is -114.1332879317e-3 ppm.

## FETCh:WLAN:AY:SUMMary:FERRor:AVERage:STATus? (Query Only)

Query the Average Frequency Error status in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:FERRor:AVERage:STATus?

### Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:FERR:AVER:STAT? may return PASS, FAIL, or NA.

## **FETCh:WLAN:AY:SUMMary:FERRor:HERTz? (Query Only)**

Query the Frequency Error value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:FERRor:HERTz?

### **Returns**

<NRf>

Units: Hz or kHz

### **Examples**

FETC:WLAN:AY:SUMM:FERR:HERT? may return 2.875e+3, which indicates that the Frequency Error value is 2.875 kHz.

## **FETCh:WLAN:AY:SUMMary:FERRor:MAX? (Query Only)**

Query the Maximum Frequency Error value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:FERRor:MAX?

### **Returns**

<NRf>

Units: ppm

### **Examples**

FETC:WLAN:AY:SUMM:FERR:MAX? may return -114.1332879317e-3, which indicates that the Maximum Frequency Error is -114.1332879317e-3 ppm.

## **FETCh:WLAN:AY:SUMMary:FERRor:PPM? (Query Only)**

Query the Frequency Error value in the Summary display.



---

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AY:SUMMary:FERRor:PPM?`

## Returns

<NRf>

Units: ppm

## Examples

`FETC:WLAN:AY:SUMM:FERR:PPM?` may return `575.023e-3`, which indicates that the Frequency Error value is `575.023e-3` ppm.

## **FETCh:WLAN:AY:SUMMary:FERRor:STATus? (Query Only)**

Query the Frequency Error status in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

`FETCh:WLAN:AY:SUMMary:FERRor:STATus?`

## Returns

<string>

## Examples

`FETC:WLAN:AY:SUMM:FERR:STAT?` may return PASS, FAIL, or NA.

## **FETCh:WLAN:AY:SUMMary:IQ:GAIN:IMBALance? (Query Only)**

Query the IQ Gain Imbalance value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:IQ:GAIN:IMBALance?

## Returns

<NRf>

Units: dB

## Examples

FETC:WLAN:AY:SUMM:IQ:GAIN:IMBAL? may return 69.2700818181e-3, which indicates that the IQ Gain Imbalance value is 69.2700818181e-3 dB.

## FETCh:WLAN:AY:SUMMary:IQ:ORIGin:OFFSet? (Query Only)

Query the IQ Origin Offset value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:IQ:ORIGin:OFFSet?

## Returns

<NRf>

Units: dB

## Examples

FETC:WLAN:AY:SUMM:IQ:ORIG:OFFS? may return -160.0000000000, which indicates that the IQ Origin Offset value is -160.0000000000 dB.

## FETCh:WLAN:AY:SUMMary:IQ:QUADrature:ERROr? (Query Only)

Query the IQ Quadrature Error value (degrees) in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:IQ:QUADrature:ERROr?
```

## Returns

<NRf>

Units: degrees

## Examples

FETC:WLAN:AY:SUMM:IQ:QUAD:ERRO? may return -220.5967903137e-3, which indicates that the IQ Quadrature Error value is -220.5967903137e-3° degrees.

## FETCh:WLAN:AY:SUMMary:PACKet:AGGRegation? (Query Only)

Query the value of the Aggregation packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:PACKet:AGGRegation?
```

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:AGGR? may return 1111, which indicates that the Aggregation value is 1111.

## FETCh:WLAN:AY:SUMMary:PACKet:BEAMformed? (Query Only)

Query the value of the Beamformed packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:PACKet:BEAMformed?

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:BEAM? may return 0, which indicates that the value of the Beamformed packet is 0.

## FETCh:WLAN:AY:SUMMary:PACKet:BWBitmap? (Query Only)

Query the value of the Bandwidth BitMap packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:PACKet:BWBitmap?

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:BWb? may return 11000000, which indicates that the value of the Bandwidth BitMap packet is 11000000.

## FETCh:WLAN:AY:SUMMary:PACKet:COMPbw? (Query Only)

Query the value of the Compressed BW packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCh:WLAN:AY:SUMMary:PACKet:COMPbw?

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:COMP? may return 2, which indicates that the value of the Compressed BW packet is 2.

## FETCh:WLAN:AY:SUMMary:PACKet:CRC? (Query Only)

Query the value of the CRC packet information in the Summary display.

### Conditions

Measurement view 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:PACKet:CRC?

### Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:CRC? may return 0110001001011010, which indicates that the value of the CRC packet is 0110001001011010.

## FETCh:WLAN:AY:SUMMary:PACKet:EDMGmcs? (Query Only)

Query the value of the EDMG MCS packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:PACKet:EDMGmcs?

### Returns

<NRf>

## Examples

FETC:WLAN:AY:SUMM:PACK:EDMG? may return 5, which indicates that the value of the EDMG MCS packet is 5.

## **FETCh:WLAN:AY:SUMMary:PACKet:LDPCcwlLen? (Query Only)**

Query the value of the additional LPDC CW Length packet information in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:PACKet:LDPCcwlLen?

### **Returns**

<string>

### **Examples**

FETC:WLAN:AY:SUMM:PACK:LDPC? may return 0, which indicates that the value of the LDPC CW Length packet is 0.

## **FETCh:WLAN:AY:SUMMary:PACKet:PRIMarychannel? (Query Only)**

Query the value of the Primary Channel packet information in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:PACKet:PRIMarychannel?

### **Returns**

<string>

### **Examples**

FETC:WLAN:AY:SUMM:PACK:PRIM? may return 0, which indicates that the value of the Primary Channel packet is 0.

## **FETCh:WLAN:AY:SUMMary:PACKet:PSDUlen? (Query Only)**

Query the Length value of the packet information in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:PACKet:PSDUlen?
```

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:PSDU? may return 1000, which indicates that the Length value of the packet is 1000.

## FETCh:WLAN:AY:SUMMary:PACKet:RSSI? (Query Only)

Query the value of the last RSSI packet information in the Summary display.

The first 2 bits of the last RSSI will fetch the Guard Interval type for EDGM (00/01/10 - Short/Normal/Long).

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:AY:SUMMary:PACKet:RSSI?
```

## Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PACK:RSSI? may return 0000, which indicates that the last value of the RSSI packet is 0000.

## FETCh:WLAN:AY:SUMMary:PACKet:SCRAMbler:INITialization? (Query Only)

Query the value of the Scrambler Initialization packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCH:WLAN:AY:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION?

## Returns

<string>

## Examples

FETCH:WLAN:AY:SUMMARY:PACKET:SCRAMBLER:INIT? may return 0101110, which indicates that the value of the Scrambler Initialization packet is 0101110.

## FETCH:WLAN:AY:SUMMARY:PACKET:SPATIALSTREAMS? (Query Only)

Query the value of the Spatial Streams packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCH:WLAN:AY:SUMMARY:PACKET:SPATIALSTREAMS?

## Returns

<string>

## Examples

FETCH:WLAN:AY:SUMMARY:PACKET:SPAT? may return 0, which indicates that the value of the Spatial Streams packet is 0.

## FETCH:WLAN:AY:SUMMARY:PACKET:STBCAPPLIED? (Query Only)

Query the value of the STBC Applied packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Fetch commands

## Syntax

FETCH:WLAN:AY:SUMMARY:PACKET:STBCAPPLIED?

## Returns

<string>



## Examples

FETC:WLAN:AY:SUMM:PACK:STBC? may return 0, which indicates that the value of the STBC Applied packet is 0.

## FETCh:WLAN:AY:SUMMary:PACKet:TRNLen ? (Query Only)

Query the value of the EDGM TRN Length packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:PACKet:TRNLen ?

### Returns

<string>

## Examples

FETC:WLAN:AY:SUMM:PAc:TRNL? may return 0, which indicates that the value of the EDGM TRN Length packet is 0.

## FETCh:WLAN:AY:SUMMary:RCPI? (Query Only)

Query the RCPI (Received Channel Power Indicator) value in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Fetch commands

### Syntax

FETCh:WLAN:AY:SUMMary:RCPI?

### Returns

<NR1>

## Examples

FETC:WLAN:AY:SUMM:RCPI? may return 207.0000000000, which indicates that the RCPI value is 207.0000000000.

## **FETCh:WLAN:AY:SUMMary:RF:POWer? (Query Only)**

Query the Radio Frequency (RF) Output Power value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:RF:POWer?

### **Returns**

<NRf>

You must select the units in the Units tab.

### **Examples**

FETC:WLAN:AY:SUMM:RF:POW? may return -6.3434391022, which indicates that the Radio Frequency Output Power value is -6.3434391022.

## **FETCh:WLAN:AY:SUMMary:SIG:QUAlity? (Query Only)**

Query the Signal Quality value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:AY:SUMMary:SIG:QUAlity?

### **Returns**

<NRf>

Units: rho

### **Examples**

FETC:WLAN:AY:SUMM:SIG:QUA? may return 999.6323e-3, which indicates that the Signal Quality value is 999.6323e-3 rho.

## **FETCh:WLAN:CONSte:MAGNitude? (Query Only)**

Returns the magnitude data for the WLAN Constellation measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:CONStE:MAGNitude?
```

## Returns

Returned data is in the form

#<num\_digit><num\_byte><data(1)><data(2)>..<data(n)> where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data(n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

FETCH:WLAN:CONS:MAGN? might return #510400.. (10400-byte data) for the measurement.

## FETCH:WLAN:CONStE:PHASe? (Query Only)

Returns the phase values for the WLAN Constellation measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:CONStE:PHASe?
```

## Returns

Returned data is in the form

#<num\_digit><num\_byte><data(1)><data(2)>..<data(n)>

where

- <num\_digit> is the number of digits in <num\_byte>
- <num\_byte> is the number of bytes of data that follow
- <data(n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

`FETCH:WLAN:CONS:PHAS?` might return #510400.. (10400-byte data) for the measurement.

## FETCh:WLAN:CONSte:TYPE? (Query Only)

Returns the type (context) for the WLAN Constellation measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Fetch commands

### Syntax

`FETCh:WLAN:CONSte:TYPE?`

### Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the nth data point in 16-bit little endian integers. The integers must be translated to the proper context as listed below.

Integer value Type (context):

- 1 Pilot
- 2 Data

## Examples

`FETCH:WLAN:CONS:TYPE?` might return #41352.. (1352-byte data) for the measurement. The actual data must be decoded to the context type using the table above.

## FETCh:WLAN:CONSte:VALue? (Query Only)

Returns the value (decoded symbol) for the WLAN Constellation measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Fetch commands

## Syntax

```
FEtCh:WLAN:CONStE:VALue?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers.

## Examples

FEtCh:WLAN:CONStE:VALue? might return #41352 indicating 1352 bytes of data for the measurement.

## FEtCh:WLAN:CRESpOse:MAGNitude? (Query Only)

Returns the magnitude data of the WLAN Channel Response measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FEtCh:WLAN:CRESpOse:MAGNitude?
```

## Returns

Returned data is in the form

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data(n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

FEtCh:WLAN:CRESpOse:MAGN? might return #510400.. (10400-byte data) for the measurement.

## FEtCh:WLAN:CRESpOse:PHASe? (Query Only)

Returns the phase data of the WLAN Channel Response measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:CRESPonse:PHASe?
```

## Returns

Returned data is in the form

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

where

- <num\_digit> is the number of digits in <num\_byte>
- <num\_byte> is the number of bytes of data that follow
- <data(n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

FETCh:WLAN:CRESP:PHAS? might return #510400.. (10400-byte data) representing the phase data for the measurement.

## FETCh:WLAN:EVM:TRACe<x>? (Query Only)

Returns the trace data for the selected WLAN EVM trace. When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol. When <x> is 3, the parameter is Average versus Subcarrier

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:EVM:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:WLAN:EVM:TRAC2?` might return #510400.. (10400-byte data) for the measurement.

## FETCh:WLAN:FLATness:PASS? (Query Only)

Queries whether the average power level values of the subcarriers across the signal bandwidth on the WLAN Spectral Flatness display remain within the limits defined for a particular standard.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:FLATness:PASS?`

## Returns

Returns "1" in the case that all data remains within the defined limits, and "0" otherwise.

## Examples

`FETCH:WLAN:FLAT:PASS?` might return 1, which means that the average power level values of the subcarriers across the signal bandwidth remain within the defined limits.

## FETCh:WLAN:FLATness:RESult? (Query Only)

Returns the summary results of each segment of the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:FLATness:RESult?`

## Returns

The data is returned as 12 comma delineated text values. The specific values are :

- minimum subcarrier within segment 1
- minimum subcarrier average energy deviation from the global average within segment 1
- deviation the of minimum subcarrier average from the allowed value in segment 1
- maximum subcarrier within segment 1
- maximum subcarrier average energy deviation from the global average within segment 1
- deviation of the maximum subcarrier average from the allowed value in segment 1
- minimum subcarrier within segment 2
- minimum subcarrier average energy deviation from the global average within segment 2
- deviation the of minimum subcarrier average from the allowed value in segment 2
- maximum subcarrier within segment 2
- maximum subcarrier average energy deviation from the global average within segment 2
- deviation of the maximum subcarrier average from the allowed value in segment 2

## Examples

`FETCH:WLAN:FLAT:RES?` might return `-6.0000000000,-17.0626174659E-3,3.9829373825,12.0000000000,19.979260` which represents the summary results of each segment of the WLAN Spectral Flatness display.

## FETCh:WLAN:FLATness:TRACe<x>? (Query Only)

Returns the WLAN Spectral Flatness trace data. When <x> is 1, the parameter is Matrix (symbol deviation per subcarrier). When <x> is 2, the parameter is Average Deviation vs Subcarrier.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:FLATness:TRACe<x>?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.



<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:WLAN:FLAT:TRACE2?` might return #510400.. (10400-byte data) for the measurement.

## **FETCH:WLAN:MERRor:TRACe<x>? (Query Only)**

Returns the trace data for the selected WLAN Magnitude Error trace.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCH:WLAN:MERRor:TRACe<x>?`

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>..<data(n)>

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:WLAN:MERR:TRAC2?` might return #510400.. (10400-byte data) for the measurement.

## **FETCH:WLAN:PERRor:TRACe<x>? (Query Only)**

Returns the trace data for the selected WLAN Phase Error trace.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:PERRor:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

```
FETCH:WLAN:PERR:TRAC2? might return #510400.. (10400-byte data) for the measurement.
```

## FETCH:WLAN:PVTime:BURSt:POWer? (Query Only)

Returns the average power of the burst packet, in dBm, for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:PVTime:BURSt:POWer?
```

## Returns

Floating point number that represents the average power of the burst packet, in dBm.

## Examples

```
FETCH:WLAN:PVT:BURST:POWER? might return -1.99, which represents the value of the average power of the burst packet of -1.99 dBm.
```

## FETCH:WLAN:PVTime:BURSt:WIDTh? (Query Only)

Returns the value of the measured time width of the burst packet from Power-On to Power-Down (or end of waveform, if that occurs before Power-Down), in seconds, for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:PVTime:BURSt:WIDTh?
```

## Returns

Floating point number which represents the value of the measured time width of the burst packet from Power-On to Power-Down (or end of waveform, if that occurs before Power-Down), in seconds.

## Examples

FETCH:WLAN:PVT:BURST:WIDTH? might return 156.0000000000E-6, which represents a width of the burst packet of 156  $\mu$ S.

## FETCH:WLAN:PVTime:FALL:TIME? (Query Only)

Returns the value of the width of the falling edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCH:WLAN:PVTime:FALL:TIME?
```

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

FETCH:WLAN:PVT:FALL:TIME? might return 7e-6, which indicates the width of the falling edge is 7  $\mu$ s.

## FETCH:WLAN:PVTime:FALL:TIME:START? (Query Only)

Returns the value of the start time of the falling edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:PVTime:FALL:TIME:START?
```

## Returns

Floating point number which represents the value of the start time of the falling edge of the burst, in seconds.

## Examples

FETCh:WLAN:PVT:FALL:TIME:STAR? might return 6e-6, which indicates that the start time of the falling edge is at 6  $\mu$ s.

## FETCh:WLAN:PVTime:FALL:TIME:STOP? (Query Only)

Returns the value of the stop time of the falling edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:PVTime:FALL:TIME:STOP?
```

## Returns

Floating point number which represents the value of the stop time of the falling edge of the burst, in seconds.

## Examples

FETCh:WLAN:PVT:FALL:TIME:STOP? might return 7e-6, which indicates the stop time of the falling edge is at 7  $\mu$ s.

## FETCh:WLAN:PVTime:FTIME? (Query Only)

Returns the value of the width of the falling edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:PVTime:FTIME?
```

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

FETCh:WLAN:PVT:FALL:TIME? might return 7e-6, which indicates the width of the falling edge is 7  $\mu$ s.

## FETCh:WLAN:PVTime:RISE:TIME? (Query Only)

Returns the value of the width of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:PVTime:RISE:TIME?
```

## Returns

Floating point number which represents the value of the width of the rising edge of the burst, in seconds.

## Examples

FETCh:WLAN:PVT:RISE:TIME? might return 7e-6, which indicates the width of the rising edge is 7  $\mu$ s.

## FETCh:WLAN:PVTime:RISE:TIME:START? (Query Only)

Returns the value of the start time of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:RISE:TIME:STARt?`

## Returns

Floating point number which represents the value of the start time of the rising edge of the burst, in seconds.

## Examples

`FETCh:WLAN:PVT:RISE:TIME:STAR?` might return `6e-6`, which indicates the start time of the rising edge is at  $6\ \mu\text{s}$ .

## FETCh:WLAN:PVTime:RISE:TIME:STOP? (Query Only)

Returns the value of the stop time of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:RISE:TIME:STOP?`

## Returns

Floating point number which represents the value of the stop time of the rising edge of the burst, in seconds.

## Examples

`FETCh:WLAN:PVT:RISE:TIME:STOP?` might return `7e-6`, which indicates the stop time of the rising edge is at  $7\ \mu\text{s}$ .

## FETCh:WLAN:PVTime:RTIME? (Query Only)

Returns the value of the width of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:RTIME?`

## Returns

Floating point number which represents the value of the width of the rising edge of the burst, in seconds.

## Examples

`FETCH:WLAN:PVT:RISE:TIME?` might return `7e-6`, which indicates the width of the rising edge is 7  $\mu$ s.

## FETCh:WLAN:PVTime:TRACe:X? (Query Only)

Returns the horizontal values (time in seconds) for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:TRACe:X?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

`<num_digit>` the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCH:WLAN:PVT:TRAC:X` might return `#43204` (3204-byte data), which represents the time in seconds (horizontal values) for the WLAN Power vs. Time trace.

## FETCh:WLAN:PVTime:TRACe:XY? (Query Only)

Returns the horizontal value (time in seconds) and vertical value (power) for the WLAN Power vs. Time trace.

`<x(n)><y(n)>` is the horizontal value and vertical value pair at the nth data point.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:TRACe:XY?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:WLAN:PVT:TRAC:XY?` might return `#43204` (3204-byte data), which represents the horizontal value (time in seconds) and vertical value (power) pair at the `n`th data point.

## FETCh:WLAN:PVTime:TRACe[:Y]? (Query Only)

Returns the vertical values (power) for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

`FETCh:WLAN:PVTime:TRACe[:Y]?`

## Returns

`#<num_digit><num_byte><data(1)><data(2)>..<data(n)>`

Where:

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the data value at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`FETCh:WLAN:PVT:TRAC:Y?` might return `#43204` (3204-byte data), which represents the vertical values (power) for the WLAN Power vs. Time trace.

## FETCh:WLAN:STABLE:VALUe? (Query Only)

Returns the number of bytes in the symbol table for the WLAN measurement.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:STABLe:VALUe?
```

## Returns

```
#<num_digit><num_byte>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

## Examples

FETCh:WLAN:STABLE:VALUE? might return #43848 indicating that there are 3848 bytes of data in the symbol table.

## FETCh:WLAN:SUMMary:BURSt:INDex? (Query Only)

Returns the index of the analyzed packet within the analysis record.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:BURSt:INDex?
```

## Arguments

None

## Returns

```
<NRf>
```

## FETCh:WLAN:SUMMary:BURSt:POWer? (Query Only)

Returns the average power of all symbols in the packet, including Preamble and Data segments.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:BURSt:POWer?
```

## Arguments

None

## Returns

<NRf> data in dBm.

## FETCh:WLAN:SUMMary:BURSt:POWer:CFACTOR? (Query Only)

Returns the peak-to-average burst power factor in the WLAN measurement. The ratio of the highest instantaneous signal power level to the average signal power.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:BURSt:POWer:CFACTOR?
```

## Arguments

None

## Returns

<NRf> data in dB.

## FETCh:WLAN:SUMMary:CERRor? (Query Only)

Returns the symbol clock error in the WLAN measurement. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:CERRor?
```

## Arguments

None

## Returns

<NRf> data in parts per million.

## FETCh:WLAN:SUMMary:CPE? (Query Only)

Returns the RMS magnitude error of the pilots over all data symbols.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:CPE?
```

## Arguments

None

## Returns

<NRf> data in percent.

## FETCh:WLAN:SUMMary:EVM:PEAK:ALL? (Query Only)

Returns the peak EVM value for all subcarriers in the WLAN measurement or for all chips depending on the selected standard.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:PEAK:ALL?

## Arguments

None

## Returns

<NRf> data in dB.

## FETCh:WLAN:SUMMary:EVM:PEAK:CHIP:ALL? (Query Only)

Returns the chip at which the reported peak EVM value occurred. Available only for 802.11b

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:PEAK:CHIP:ALL?

## Arguments

None

## Returns

<NR1>

## FETCh:WLAN:SUMMary:EVM:PEAK:CHIP:ONEKchips? (Query Only)

Returns the chip at which the reported peak EVM value over the first 1000 chips occurred.

Only available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

---

## Syntax

FETCh:WLAN:SUMMary:EVM:PEAK:CHIP:ONEKchips?

## Arguments

None

## Returns

<NR1>

## FETCh:WLAN:SUMMary:EVM:PEAK:DATA? (Query Only)

Returns the peak EVM value for the data subcarriers in the WLAN measurement. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:PEAK:DATA?

## Arguments

None

## Returns

<NRf> data in dB.

## FETCh:WLAN:SUMMary:EVM:PEAK:ONEKchips? (Query Only)

Returns the peak EVM value measured over the first 1000 chips. Available only for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:PEAK:ONEKchips?

## Arguments

None

## Returns

<NRf> data in dB.

## **FETCh:WLAN:SUMMary:EVM:PEAK:PILots? (Query Only)**

Returns the peak EVM value for the pilots in the WLAN measurement. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:PEAK:PILots?
```

## Arguments

None

## Returns

<NRf> data in dB.

## **FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:ALL? (Query Only)**

Returns the subcarrier on which the ALL peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:ALL?
```

## Arguments

None

---

## Returns

<NR1>

## FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:DATA? (Query Only)

Returns the subcarrier on which the DATA peak EVM value occurred. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:DATA?
```

## Arguments

None

## Returns

<NR1>

## FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:PILOts? (Query Only)

Returns the subcarrier on which the PILOT peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:PEAK:SCARrier:PILOts?
```

## Arguments

None

## Returns

<NR1>

## **FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:ALL? (Query Only)**

Returns the symbol on which the reported EVM value occurred.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:ALL?

### **Arguments**

None

### **Returns**

<NR1>

## **FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:DATA? (Query Only)**

Returns the symbol on which the DATA peak EVM value occurred. Not available for 802.11b.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:DATA?

### **Arguments**

None

### **Returns**

<NR1>

## **FETCh:WLAN:SUMMary:EVM:PEAK:SYMBol:PILOts? (Query Only)**

Returns the symbol on which the PILOT peak EVM value occurred.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:PEAK:SYMBOL:PILOTS?
```

## Arguments

None

## Returns

<NR1>

## FETCh:WLAN:SUMMary:EVM:RMS:ALL? (Query Only)

Returns the RMS EVM data for all subcarriers in the WLAN measurement and for all chips for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:RMS:ALL?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## FETCh:WLAN:SUMMary:EVM:RMS:AVERAge? (Query Only)

Returns the average RMS EVM value in the WLAN measurement over the selected number of bursts.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:RMS:AVERAge?

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## FETCh:WLAN:SUMMary:EVM:RMS:DATA? (Query Only)

Returns the RMS EVM data for the data subcarriers in the WLAN measurement. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

FETCh:WLAN:SUMMary:EVM:RMS:DATA?

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## FETCh:WLAN:SUMMary:EVM:RMS:MAXimum? (Query Only)

Returns the maximum RMS EVM value in the WLAN measurement over the selected number of bursts.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

---

## Syntax

```
FETCh:WLAN:SUMMary:EVM:RMS:MAXimum?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## FETCh:WLAN:SUMMary:EVM:RMS:ONEKchips? (Query Only)

Returns the EVM RMSdata for the 1k chips in the WLAN measurement measured over the first 1000 chips.

Only available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:RMS:ONEKchips?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## FETCh:WLAN:SUMMary:EVM:RMS:PILots? (Query Only)

Returns the RMS EVM data for the pilot subcarriers in the WLAN measurement. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:EVM:RMS:PILots?
```

## Arguments

None

## Returns

<NRf> data in percent.

## **FETCh:WLAN:SUMMary:FERRor? (Query Only)**

Returns the frequency difference between the measured carrier frequency of the signal and the measurement frequency setting.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:FERRor?
```

## Arguments

None

## Returns

<NRf> data in kHz.

## **FETCh:WLAN:SUMMary:HEADer? (Query Only)**

Returns the received packet header information for the WLAN measurement. Applies to the 802.11b standard only.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:HEADer?
```

## Arguments

None

**Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

**FETCh:WLAN:SUMMary:HTSig? (Query Only)**

Returns the decimal values of the received packet HT-SIGNAL. Applies to the 802.11n standard only.

**Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

**Group**

Fetch commands

**Syntax**

```
FETCh:WLAN:SUMMary:HTSig?
```

**Arguments**

None

**Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

**FETCh:WLAN:SUMMary:IQOffset? (Query Only)**

Returns the average magnitude of the DC subcarrier level relative to total signal power.

**Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

**Group**

Fetch commands

**Syntax**

```
FETCh:WLAN:SUMMary:IQOffset?
```

**Arguments**

None

**Returns**

<NRf> data in dB.

## **FETCh:WLAN:SUMMary:LSIG? (Query Only)**

Returns the decimal values of the received packet (legacy) SIGNAL symbols fields. Applies to the 802.11n standard only.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:SUMMary:LSIG?

### **Arguments**

None

### **Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

## **FETCh:WLAN:SUMMary:PACKet? (Query Only)**

Returns the type and number of symbols, EVM-RMS and average power of the Preamble and Data portions of the packet.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Fetch commands

### **Syntax**

FETCh:WLAN:SUMMary:PACKet?

### **Arguments**

None

### **Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

## **FETCh:WLAN:SUMMary:PACKet:DMODulation? (Query Only)**

Returns the modulation used in the Data symbols.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:PACKet:DMODulation?
```

## Returns

<string> The data modulation being used.

## FETCh:WLAN:SUMMary:PACKet:FORMat? (Query Only)

Returns the packet format setting for the WLAN measurement

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:PACKet:FORMat?
```

## Arguments

None

## Returns

<string> The packet format setting.

## FETCh:WLAN:SUMMary:PACKet:GUARd? (Query Only)

Returns the packet guard interval used by the DATA symbols. Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:PACKet:GUARd?
```

## Arguments

None

## Returns

<NRf> The packet guard interval, 1/4 or 1/8.

## **FETCh:WLAN:SUMMary:SIG? (Query Only)**

Returns the decimal values of the received packet SIGNAL symbols fields. Available only for 802.11agjp.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:SIG?
```

## Arguments

None

## Returns

Results are returned in the same left-to-right order in which they are presented in the display.

## **FETCh:WLAN:SUMMary:VHTSig:A? (Query Only)**

Returns the decimal values of the received packet VHT-SIGNAL(A) symbols fields.

Available only for 802.11ac.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Fetch commands

## Syntax

```
FETCh:WLAN:SUMMary:VHTSig:A?
```

## Arguments

None



**Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

**FETCh:WLAN:SUMMary:VHTSig:B? (Query Only)**

Returns the decimal values of the received packet VHT-SIGNAL(B) symbols fields.

Available only for 802.11ac.

**Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

**Group**

Fetch commands

**Syntax**

```
FETCh:WLAN:SUMMary:VHTSig:B?
```

**Arguments**

None

**Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

**\*IDN? (Query Only)**

Returns the analyzer identification code.

**Conditions**

Measurement views: All

**Group**

IEEE common commands

**Syntax**

```
*IDN?
```

**Arguments**

None

**Returns**

The analyzer identification code in the following format

```
TEKTRONIX,RSAXXXXX,<serial_number>,<firmware_version>
```

Where

TEKTRONIX indicates that the manufacturer is Tektronix.

RSAXXXXX is the model number (RSA5103B, RSA5106B, RSA5115B, or RSA5126B).

<serial\_number> is the serial number.

<firmware\_version> is the firmware version.

## Examples

\*IDN? might return the response

```
TEKTRONIX,RSA5126B,B000111,FV2.6.1500.
```

## INITiate:CONTinuous

Determines whether to place the analyzer in the single or the continuous acquisition mode.

### Conditions

Measurement views: All

### Group

Initiate commands

### Syntax

```
INITiate:CONTinuous { OFF | ON | 0 | 1 }
```

```
INITiate:CONTinuous?
```

### Related Commands

```
INITiate:RESumeINITiate[:IMMediate]
```

### Arguments

OFF or 0 places the analyzer in the single acquisition mode. To initiate the acquisition, use the `INITiate[:IMMediate]` command. To stop acquisition when a trigger is not present, and to avoid a 2.7 hour time-out, send the free-run trigger command, `TRIGger:SEQuence:STATus 0`. Then, send the command `INITiate:CONTinuous OFF`. This sequence ensures a minimal delay before the `INIT:CONT OFF` command completes.

ON or 1 places the analyzer in the continuous acquisition mode. To initiate the acquisition, use the `INITiate[:IMMediate]` command. To stop the acquisition in the continuous mode, send the following command: `INITiate:CONTinuous OFF`

### Returns

See Arguments.

### Examples

```
INITIATE:CONTINUOUS ON places the analyzer in the continuous acquisition mode.
```

## INITiate[:IMMediate] (No Query Form)

Starts input signal acquisition.



**Note:** *It is an overlapped command, which does not finish executing before the next command starts executing. Use the \*OPC(?) and \*WAI commands to synchronize all pending operations to the execution of this command.*

### Conditions

Measurement views: All

### Group

Initiate commands

### Syntax

```
INITiate[:IMMediate]
```

### Related Commands

\*OPC, \*TRG, \*WAI, INITiate:CONTinuous

### Arguments

None

### Examples

INITIATE:IMMEDIATE starts input signal acquisition.

## INITiate:RESume (No Query Form)

Restarts signal processing.

Resume is only applicable when FastFrame is enabled, and when the instrument is stopped while analyzing the frames. When the instrument is stopped, measurements will finish processing and the system might stop before all of the frames of an acquisition are processed (when acquisition is stopped, the system stops as soon as all measurements say they are finished processing the current acquisition). However, the instrument saves all of the frames to the Acquisition History (Data Cache), and when INITiate:RESume is sent, it will begin processing at the first frame of the last acquisition that was not processed (the one after the last frame processed), and process all of these remaining frames before starting a new acquisition. If INITiate is sent instead, any remaining frames are discarded and a new acquisition is started immediately. If FastFrame is off, or there are no remainder frames, or the acquisition was aborted rather than stopped, then INITiate:RESume and INITiate are the same.

### Conditions

Measurement views: All

FastFrame enabled

### Group

Initiate commands

## Syntax

INITiate:RESume

## Related Commands

\*OPC, \*TRG, \*WAI, INITiate:CONTinuousINITiate[:IMMediate]

## Arguments

None

## Examples

INITIATE:RESUME restarts input signal acquisition.

# Input commands

## INPut:CORRection:EXTErnal:EDIT<x>:INTErpolation

Sets or queries the interpolation setting to use with the indicated external gain table.

### Conditions

Measurement views: All

### Group

Input commands

### Syntax

INPut:CORRection:EXTErnal:EDIT<x>:INTErpolation { LINear | LOGarithmic }

INPut:CORRection:EXTErnal:EDIT<x>:INTErpolation

### Related Commands

INPut:CORRection:EXTErnal:EDIT<x>:NEW

### Arguments

LINear select when the frequency scale of the spectrum or spurious measurements are linear.

LOGarithmic select when the frequency scale of the spectrum is logarithmic.

### Examples

INPut:CORRection:EXTErnal:EDIT1:INTErpolation LINear uses linear interpolation when applying External correction Table one (1) to the chosen waveforms.

## INPut:CORRection:EXTErnal:EDIT<x>:LABel

Sets or queries the name of the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.



**Note:** This command has replaced

`CALibration:CORRection:EXTernal:EDIT<x>:LABEL`

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut:CORRection:EXTernal:EDIT<x>:LABel <name>
```

```
INPut:CORRection:EXTernal:EDIT<x>:LABel?
```

## Arguments

<name>::=<string> specifies the name of the external loss table.

## Examples

```
INPut:CORRection:EXTernal:EDIT<x>:LABel <name> 'Sample Table 1' names the External Loss Table 1 "Sample Table 1".
```

## INPut:CORRection:EXTernal:EDIT<x>:NEW

Creates the indicated external loss table from a series of frequency and gain value pairs.



**Note:** This command has replaced `CALibration:CORRection:EXTernal:EDIT<x>:NEW`

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut:CORRection:EXTernal:EDIT<x>:NEW<freq(1)>,<gain(1)>,<freq(2)>,<gain(2)>,...,<freq(n)>,<gain(n)>
```

```
INPut:CORRection:EXTernal:EDIT<x>:NEW?
```

## Related Commands

```
INPut:CORRection:EXTernal:TYPE
```

## Arguments

<freq(n)>,<gain(n)> specifies a pair of frequency (<NR3> in Hz) and gain (+ | -<NR3> in dB) in the external correction table.



**Note:** Do not include a unit suffix (such as Hz, dBm, or dB) with the arguments for this command.

The setting range is:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

Loss: -50 to +30 dB. Negative values indicate loss and positive values indicate gain. Resolution is 0.1 dB.

## Examples

`INPut:CORRection:EXTErnal:EDIT?:NEW 1.0E+9,-2.2,-1.5E+9,-2.3` creates the External correction Table specifying a loss of 2.2 dB at 1 GHz and a loss of 2.3 dB at 1.5 GHz.

## INPut:CORRection:EXTErnal:EDIT<x>:STATe

Determines whether to enable or disable the external loss table.

The parameter <x> = 1 to 3 represent the External Loss Table 1 to 3, respectively.



**Note:** This command has replaced `CALibration:CORRection:EXTErnal:EDIT<x>:STATe`

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut:CORRection:EXTErnal:EDIT<x>:STATe { OFF | ON | 0 | 1 }
```

```
INPut:CORRection:EXTErnal:EDIT<x>:STATe?
```

## Arguments

OFF or 0 disables the external loss table.

ON or 1 enables the external loss table.



**Note:** You can enable one or more tables at the same time.

## Examples

`INPut:CORRection:EXTErnal:EDIT3:STATe? ON` enables the External Loss Table 3.

## INPut:CORRection:EXTernal:GAIN[:MAGNitude]

Sets or queries the external gain value. It can be enabled or disabled using the `INPut:CORRection:EXTernal:GAIN:STATe` command.



**Note:** This command has replaced `CALibration:CORRection:EXTernal:GAIN[:MAGNitude]`

### Conditions

Measurement views: All

### Group

Input commands

### Syntax

```
INPut:CORRection:EXTernal:GAIN[:MAGNitude] <value>
```

```
INPut:CORRection:EXTernal:GAIN[:MAGNitude]?
```

### Arguments

<value>::=<NRf> specifies the external gain value. Range: -50 to +30 dB.

### Examples

```
INPut:CORRection:EXTernal:GAIN[:MAGNitude] <value> -10 sets the external gain to -10 dB.
```

## INPut:CORRection:EXTernal:GAIN:STATe

Determines whether to enable or disable the external gain value.



**Note:** This command has replaced `CALibration:CORRection:EXTernal:GAIN:STATe`.

### Conditions

Measurement views: All

### Group

Input commands

### Syntax

```
INPut:CORRection:EXTernal:GAIN:STATe { OFF | ON | 0 | 1 }
```

```
INPut:CORRection:EXTernal:GAIN:STATe?
```

### Related Commands

```
INPut:CORRection:EXTernal:GAIN[:MAGNitude]
```

## Arguments

OFF or 0 disables the external gain value.

ON or 1 enables the external gain value.

## Examples

`INPut:CORRection:EXTErnal:GAIN:STATE ON` enables the external gain value.

## INPut:CORRection:EXTErnal:TYPE

Sets or queries the data type to use when applying the external loss table corrections.



**Note:** This command has replaced `CALibration:CORRection:EXTErnal:TYPE`

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut:CORRection:EXTErnal:TYPE { TRACe | DATA }
```

```
INPut:CORRection:EXTErnal:TYPE
```

## Related Commands

```
INPut:CORRection:EXTErnal:EDIT<x>:NEW
```

## Arguments

TRACe selects traces in the Spectrum, Spectrogram, Spurious, and Amplitude versus Time views. Selecting TRACe disables the `INPut:CORRection:EXTErnal:GAIN:STATE` command.

DATA selects all acquired data. Selecting DATA enables the `INPut:CORRection:EXTErnal:GAIN:STATE` command.

## Examples

`INPUT:CORRECTION:EXTERNAL:TYPE`TRACe selects traces to apply the external loss table corrections.

## INPut:{MLEVel|RLEVel}

Sets or queries the reference level.

## Conditions

Measurement views: All

## Group

Input commands



## Syntax

```
INPut:{MLEVel|RLEVel} <value>
```

```
INPut:{MLEVel|RLEVel}?
```

## Arguments

<value>::<NRf> specifies the reference level. Range: -170 to +50 dBm.

## Examples

```
INPUT:RLEVEL -10 sets the reference level to -10 dBm.
```

## INPut[:RF]:ATTenuation

Sets or queries the input attenuation. Programming a specified attenuation sets `INPut[:RF]:ATTenuation:AUTO OFF`.

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut[:RF]:ATTenuation <value>
```

```
INPut[:RF]:ATTenuation?
```

## Related Commands

```
INPut[:RF]:ATTenuation:AUTO
```

## Arguments

<value>::<NR1> specifies the input attenuation.

Range: 0 to 75 dB in 5 dB steps.

## Examples

```
INPUT:RF:ATTENUATION 20 sets the input attenuation to 20 dB.
```

## INPut[:RF]:ATTenuation:AUTO

Determines whether to set the input attenuation automatically or manually.

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut[:RF]:ATTenuation:AUTO { OFF | ON | 0 | 1 }  
INPut[:RF]:ATTenuation:AUTO?
```

## Arguments

OFF or 0 specifies that the input attenuation is set manually. To set it, use the `INPut[:RF]:ATTenuation` command.  
ON or 1 specifies that the input attenuation is set automatically according to the reference level.

## Examples

`INPUT:RF:ATTENUATION:AUTO ON` specifies that the input attenuation is set automatically.

## INPut[:RF]:ATTenuation:MONitor:STATe

Determines whether to enable or disable to monitor attenuator use.

## Conditions

Measurement views: All

## Group

Input commands

## Syntax

```
INPut[:RF]:ATTenuation:MONitor:STATe { OFF | ON | 0 | 1 }  
INPut[:RF]:ATTenuation:MONitor:STATe?
```

## Arguments

OFF or 0 disables to monitor attenuator use.  
ON or 1 enables to monitor attenuator use.

## Examples

`INPUT:RF:ATTENUATION:MONITOR:STATE ON` enables to monitor attenuator use.

## INPut[:RF]:GAIN:STATe

Determines whether to enable or disable the internal preamp.

## Conditions

Measurement views: All (Option 50 or 51 only)

## Group

Input commands

## Syntax

```
INPut[:RF]:GAIN:STATe { OFF | ON | 0 | 1 }
```

```
INPut[:RF]:GAIN:STATe?
```

## Arguments

OFF or 0 disables the internal preamp.

ON or 1 enables the internal preamp.

## Examples

```
INPUT:RF:GAIN:STATE ON enables the internal preamp.
```

## INPut[:RF]:PRESelector:STATe

Sets or queries the Preselector setting. The preselector uses input filters for image suppression when the span of the instrument allows for their use.

## Conditions

Only available when a RSA7100A is connected with SignalVu-PC.

## Group

Fetch commands

## Syntax

```
INPut[:RF]:PRESelector:STATe { AUTO | ON | OFF }
```

```
INPut[:RF]:PRESelector:STATe ?
```

## Arguments

AUTO : This setting balances maximizing acquisition bandwidth with image suppression.

ON : sets the preselector to always be on, independent of span.

OFF : setting bypasses the preselector filters (except for the LPF, which is always present) when the span is greater than the allowed acquisition bandwidth.

## Returns

See Arguments

## Examples

```
INPUT[:RF]:PRESELECTOR:STATE OFF sets the preselector to OFF.
```

## Memory commands

### **MMEMory:{AM|FM|PM}:LOAD:TRACe (No Query Form)**

Loads the AM/FM/PM measurement trace data from the specified file.

#### **Conditions**

Measurement views: AM, FM, PM

#### **Group**

Mass Memory commands

#### **Syntax**

```
MMEMory:{AM|FM|PM}:LOAD:TRACe <file_name>
```

#### **Arguments**

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .AmVsTime (AM), .FmVsTime (FM), or .PmVsTime (PM). You can omit the extension.

#### **Examples**

MMEMORY:AM:LOAD:TRACE "Sample1" loads the trace data from the *Sample1.AmVsTime* file in the AM measurement.

### **MMEMory:{AM|FM|PM}:SHOW:TRACe<x>**

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

#### **Conditions**

Measurement views: AM, FM, PM

#### **Group**

Mass Memory commands

#### **Syntax**

```
MMEMory:{AM|FM|PM}:SHOW:TRACe<x> { OFF | ON | 0 | 1 } MMEMory:{AM|FM|PM}:SHOW:TRACe<x>?
```

#### **Related Commands**

[MMEMory:{AM|FM|PM}:LOAD:TRACe](#)

#### **Arguments**

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

MMEMORY:FM:SHOW:TRACE2 ON enables display of the recalled data loaded in Trace2.

## MMEMory:{AM|FM|PM}:STORE:TRACe (No Query Form)

Stores the AM/FM/PM measurement trace data in the specified file.

### Conditions

Measurement views: AM, FM, PM

### Group

Mass Memory commands

### Syntax

```
MMEMory:{AM|FM|PM}:STORE:TRACe <file_name>
```

### Arguments

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .AmVsTime (AM), .FmVsTime (FM), or .PmVsTime (PM). You can omit the extension.

### Examples

MMEMORY:AM:STORE:TRACE "Sample1" stores the trace data in the *Sample1.AmVsTime* file in the AM measurement.

## MMEMory:AUDio:LOAD:FILTer (No Query Form)

Loads a user-defined audio filter from the specified file.

### Conditions

Measurement views: Audio Spectrum

### Group

Mass Memory commands

### Syntax

```
MMEMory:AUDio:LOAD:FILTer <file_name>
```

### Related Commands

[\[SENSe\]:AUDio:FILTer:SOURce?](#)

### Arguments

<file\_name> ::= <string> specifies the filter file to load.

## Examples

MMEMORY:AUDIO:LOAD:FILTER "filter4" loads the user-defined filter4 file for the measurement.

## MMEMory:AVTime[:MEASview<y>]:LOAD:TRACe<x> (No Query Form)

Loads the Amplitude versus Time trace data from the specified file. The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

### Conditions

Measurement views: Amplitude versus Time

### Group

Mass Memory commands

### Syntax

```
MMEMory:AVTime[:MEASview<y>]:LOAD:TRACe<x> <file_name>
```

### Arguments

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .AmplVsTime. You can omit the extension.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

MMEMORY:AVTIME:MEASVIEW1:LOAD:TRACE1 "Sample1" loads Trace 1 data on Amplitude versus Time Measurement View 1 from the *Sample1.AmplVsTime* file.

## MMEMory:AVTime[:MEASview<y>]:SHOW:TRACe<x>

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

### Conditions

Measurement views: Amplitude versus Time

### Group

Mass Memory commands

### Syntax

```
MMEMory:AVTime[:MEASview<y>]:SHOW:TRACe<x> { OFF | ON | 0 | 1 }
```

```
MMEMory:AVTime[:MEASview<y>]:SHOW:TRACe<x>?
```

## Related Commands

[MMEMory:AVTime\[:MEASview<y>\]:LOAD:TRACe<x>](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

MMEMORY:AVTIME:MEASVIEW1:SHOW:TRACE2 ON enables display of the recalled data loaded in Trace2 from Amplitude versus Time Measurement View 1.

## MMEMory:AVTime[:MEASview<y>]:STORE:PLOT

Stores the Amplitude versus Time trace data in the specified file. The parameter <x> = 1 to 4; All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Mass Memory commands

## Syntax

MMEMory:AVTime[:MEASview<y>]:STORE:PLOT<file\_name>

## Arguments

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .AmplVsTime. You can omit the extension.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

MMEMORY:AVTIME:MEASVIEW1:STORE:PLOT "Sample1" stores Trace 1 data from Amplitude versus Time Measurement View 1 in the *Sample1.AmplVsTime* file.

## MMEMory:AVTime[:MEASview<y>]:STORE:PLOT:DONE

Stores the Amplitude versus Time trace data in the specified file. The parameter <x> = 1 to 4; All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:AVTime[:MEASview<y>]:STORE:PLOT:DONe <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .AmplVsTime. You can omit the extension.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
MMEMORY:AVTIME:MEASVIEW1:STORE:PLOT:DONE "Sample1" stores Trace 1 data from Amplitude versus Time Measurement View 1 in the Sample1.AmplVsTime file.
```

## MMEMemory:AVTime[:MEASview<y>]:STORE:TRACe<x> (No Query Form)

Stores the Amplitude versus Time trace data in the specified file. The parameter <x> = 1 to 4; All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:AVTime[:MEASview<y>]:STORE:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .AmplVsTime. You can omit the extension.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
MMEMORY:AVTIME:MEASVIEW1:STORE:TRACE1 "Sample1" stores Trace 1 data from Amplitude versus Time Measurement View 1 in the Sample1.AmplVsTime file.
```



## **MMEMory:CALibration:LOAD:CORRection:EXTErnal:EDIT<x> (No Query Form)**

Loads the external loss table from a specified file.

The parameter <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

### **Conditions**

Measurement views: All

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:CALibration:LOAD:CORRection:EXTErnal:EDIT<x>
```

```
<file_name>
```

### **Arguments**

<file\_name>::<string> specifies the file to load the external loss table from. The file extension is .csv. You can omit the extension.

### **Examples**

MMEMORY:CALIBRATION:LOAD:CORRECTION:EXTERNAL:EDIT1 "Table1" loads the External Loss Table 1 from the *Table1.csv* file.

## **MMEMory:CALibration:STORe:CORRection:EXTErnal:EDIT<x> (No Query Form)**

Stores the external loss table to a specified file.

The parameter <x> = 1 to 3 represent External Loss Table 1 to 3, respectively.

### **Conditions**

Measurement views: All

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:CALibration:STORe:CORRection:EXTErnal:EDIT<x><file_name>
```

### **Arguments**

<file\_name>::<string> specifies the file to store the external loss table to. The file extension is .csv. You can omit the extension.

### **Examples**

MMEMORY:CALIBRATION:STORE:CORRECTION:EXTERNAL:EDIT1 "Table1" stores the External Loss Table 1 to the *Table1.csv* file.

## **MMEMory:CCDF:LOAD:TRACe<x> (No Query Form)**

Loads the CCDF trace data from the specified file.

### **Conditions**

Measurement views: CCDF

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:CCDF:LOAD:TRACe<x> <file_name>
```

### **Arguments**

<file\_name>::

### **Examples**

MMEMORY:CCDF:LOAD:TRACE1 "Sample1" loads Trace 1 data from the *Sample1.CCDF* file.

## **MMEMory:CCDF:SHOW:TRACe<x>**

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

### **Conditions**

Measurement views: CCDF

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:CCDF:SHOW:TRACe<x> { OFF | ON | 0 | 1 } MMEMory:CCDF:SHOW:TRACe<x>?
```

### **Related Commands**

[MMEMory:CCDF:LOAD:TRACe<x>](#)

### **Arguments**

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

### **Examples**

MMEMORY:CCDF:SHOW:TRACE2 ON enables display of the recalled data loaded in Trace2.

## MMEMory:CCDF:STORe:TRACe<x> (No Query Form)

Stores the CCDF trace data in the specified file.

### Conditions

Measurement views: CCDF

### Group

Mass Memory commands

### Syntax

```
MMEMory:CCDF:STORe:TRACe<x> <file_name>
```

### Arguments

<file\_name>::

### Examples

MMEMORY:CCDF:STORE:TRACE1 "Sample1" stores Trace 1 data in the *Sample1.CCDF* file.

## MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:MEASurement:UOTHer (No query form)

Loads a user-defined measurement filter from the specified file of the measurement source.

### Group

Mass Memory commands

### Syntax

```
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:MEASurement:UOTHer <file_name>
```

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:REFerence](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:MEASurement](#)

### Arguments

<x> = 1 to 8, representing Range of Source.

<file\_name>::

### Examples

MMEMORY:DDEMOD:SOURCE1:LOAD:FILTER:MEASUREMENT:UOTHER "Table2" loads User other filter from the file Table2.csv of the measurement source 1.

## **MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:MEASurement:USER<y> (No query form)**

Loads the user-defined measurement filter from the specified file of the measurement source. The parameter <y> specifies one of the User Meas Filters 1 to 3. The filter file must have a .csv file extension, though you may omit the file extension in the command line.

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:MEASurement:USER<y> <file_name>
```

### **Related Commands**

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:REFerence](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:MEASurement](#)

### **Arguments**

<x> = 1 to 8, representing Range of Source.

<file\_name>::<string> specifies the filter file to load.

### **Examples**

```
MMEMORY:DDEMOD:SOURCE1:LOAD:FILTER:MEASUREMENT:USER1 "Table1" loads User Meas Filter 1 from the file Table1.csv of the measurement source 1.
```

## **MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFerence:UOTHer (No query form)**

Loads the specified user-defined reference filter, UOTHer, from a specified file of the measurement source.

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFerence:UOTHer<file_name>
```

### **Related Commands**

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:REFerence](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:MEASurement](#)

### **Arguments**

<x> = 1 to 8, representing Range of Source.

<file\_name>::<string> specifies the reference filter file to load. The file extension is type .csv.

## Examples

MMEMORY:DDEMOD:SOURCE1:LOAD:FILTER:REFERENCE:UOTHER "FinalFilt" loads the filter file FinalFilt.csv into the filter reference UOTHER of the measurement source 1.

## MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFeRence:USER<y> (No query form)

Loads the specified user-defined reference filter, USER1 | 2 | 3 from a specified file of the measurement source.

### Group

Mass Memory commands

### Syntax

MMEMory:DDEMod[:SOURce<x>]:LOAD:FILTer:REFeRence:USER<y> <file\_name>

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:REFeRence](#)

### Arguments

<x> = 1 to 8, representing Range of Source.

<file\_name>::<string> specifies the reference filter file to load. The file extension is type .csv.

### Examples

MMEMORY:DDEMOD:SOURCE1:LOAD:FILTER:REFERENCE:USER2 "DUTfilt2" loads the filter file DUTfilt2.csv into the filter reference USER2 of the measurement source 1.

## MMEMory:DDEMod[:SOURce<x>]:LOAD:SYMBOL:MAP (No query form)

Loads the specified symbol map filename of the measurement source. The setting correlates to the field, User Symbol Map (per Modulation Type) on the Advanced Params tab in the Digital Modulation settings.

### Group

Mass Memory commands

### Syntax

MMEMory:DDEMod[:SOURce<x>]:LOAD:SYMBOL:MAP <file\_name>

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:MODulation:TYPE](#)

### Arguments

<x> = 1 to 8, representing Range of Source.

<file\_name>::<string> specifies the symbol map file to load. The file extension is type .txt.

## Examples

`MMEMORY:DDEMOD:SOURCE1:LOAD:SYMBOL:MAP "C:\usymmap.txt"` identifies the file to load for the symbol map of the measurement source 1.

## MMEMory:DPX:LOAD:TRACe<x> (No Query Form)

Loads the DPX spectrum trace data from the specified file.

The parameter <x> = 1, 2, 3, or 5; TRACe4 (math trace) is invalid.

### Conditions

Measurement views: DPX spectrum

### Group

Mass Memory commands

### Syntax

`MMEMory:DPX:LOAD:TRACe<x> <file_name>`

### Arguments

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .dpt. You can omit the extension.

## Examples

`MMEMORY:DPX:LOAD:TRACE1 "Sample1"` loads Trace 1 data from the *Sample1.dpt* file.

## MMEMory:DPX:SHOW:TRACe<x>

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

### Conditions

Measurement views: DPX spectrum

### Group

Mass Memory commands

### Syntax

`MMEMory:DPX:SHOW:TRACe<x> { OFF | ON | 0 | 1 } MMEMory:DPX:SHOW:TRACe<x>?`

### Related Commands

[MMEMory:DPX:LOAD:TRACe<x>](#)

### Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

`MMEMORY:DPX:SHOW:TRACE2 ON` enables display of the recalled data loaded in Trace2.

## MMEMory:DPX:STORe:TRACe<x>

Stores the DPX spectrum trace data in the specified file. The parameter <x> = 1 to 5; All traces are valid.

### Conditions

Measurement views: DPX spectrum

### Group

Mass Memory commands

### Syntax

`MMEMory:DPX:STORe:TRACe<x> <file_name>`

### Arguments

`<file_name>::<string>` specifies the file in which to store trace data. The file extension is `.dpt`. You can omit the extension.

## Examples

`MMEMORY:DPX:STORE:TRACE1 "Sample1"` stores Trace 1 data in the `Sample1.dpt` file.

## MMEMory:DPX:TDM:FREQuency:TXBer:LOAD:PATtern (No Query Form)

Loads the Tx BER User data pattern from the specified file.

### Conditions

Measurement views: DPX:Frequency

### Group

Mass Memory commands

### Syntax

`MMEMory:DPX:TDM:FREQuency:TXBer:LOAD:PATtern <file_name>`

### Related Commands

`[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern`, `[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REPeat`

### Arguments

`<file_name>::<string>` that specifies the file (path) from which to load the User data pattern. The file type containing the data pattern must be in CSV format. See the instrument Help for details on creating a user pattern file.

## Examples

MMEMORY:DPX:TDM:FREQUENCY:TXBER:LOAD:PATTERN "C:\RSA5100B Files\Example Files\SampleUserPattern.csv" loads the user pattern defined in SampleUserPattern.csv.

## MMEMory:EMC:ACCEssory<x>:LOAD:TABLE (No Query Form)

Loads the loss table of the specified accessory from the specified file in the EMC view.

<x> = <NR1> 1 to 6

### Conditions

Measurement view: EMC - EMI Display

### Group

Mass Memory commands

### Syntax

MMEMory:EMC:ACCEssory<x>:LOAD:TABLE <file name>

### Arguments

<file name> = <String> specifies the file to load the loss table with extension as .emcemi

### Returns

None

## Examples

MMEMory:EMC:ACCEssory1:LOAD:TABLE "C:\SignalVu-PC Files\table.csv" loads the loss table for accessory 1 from the table .csv file.

## MMEMory:EMC:ACCEssory<x>:STORE:TABLE (No Query Form)

Stores the loss table of the specified accessory into the specified file in the EMC view.

<x> = <NR1> 1 to 6

### Conditions

Measurement view: EMC - EMI Display

### Group

Mass Memory commands

### Syntax

MMEMory:EMC:ACCEssory<x>:STORE:TABLE <file name>



## Arguments

<file name> = <String> specifies the file to store the loss table with extension as .csv

## Returns

None

## Examples

`MMEemory:EMC:ACCEssory2:STORe:TABLE "C:\SignalVu-PC Files\sample.csv"` stores the loss table for the accessory 2 in the sample.csv file.

## MMEemory:EMC:AMBIent:TRACe:FILE (No Query Form)

Recalls the ambient trace from the file in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEemory:EMC:AMBIent:TRACe:FILE <file name>`

## Arguments

<file name> = <String> specifies the file with extension as .emcemi to recall the ambient trace.

## Returns

None

## Examples

`MMEemory:EMC:AMBIent:TRACe:FILE "C:\SignalVu-PC Files\ambient.emcemi"` recalls the ambient trace from the ambient.emcemi file.

## MMEemory:EMC:AMBIent:TRACe:SHOW (No Query Form)

Sets or queries to enable or disable show recalled trace checkbox of ambient trace in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEMemory:EMC:AMBIent:TRACe:SHOW {0 | OFF | 1 | ON} MMEMemory:EMC:AMBIent:TRACe:SHOW?`

## Arguments

0 or OFF disable the show recalled trace check box of ambient trace

1 or ON enable the show recalled trace check box of ambient trace

## Returns

<Boolean> 0,1

## Examples

`MMEMemory:EMC:AMBIent:TRACe:SHOW?` might return 1, indicating that the show recalled trace check box of ambient trace is enabled.

## MMEMemory:EMC:AMBIent:TRACe:STORe (No Query Form)

Stores the ambient trace into the specified file in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEMemory:EMC:AMBIent:TRACe:STORe <file name>`

## Arguments

<file name> = <String> specifies the file to store the ambient trace with extension as .emcemi

## Returns

None

## Examples

`MMEMemory:EMC:AMBIent:TRACe:STORe "C:\SignalVu-PC Files\ambient.emcemi"` stores the ambient trace in the ambient.emcemi file.

## MMEMemory:EMC:MATH:TRACe:FILE (No Query Form)

Recalls the math trace from the file in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

```
MMEMemory:EMC:MATH:TRACe:FILE <file name>
```

## Arguments

<file name> = <String> specifies the file with extension as .emcemi to recall the math trace.

## Returns

None

## Examples

```
MMEMemory:EMC:MATH:TRACe:FILE "C:\SignalVu-PC Files\math.emcemi" recalls the math trace from the math.emcemi file.
```

## MMEMemory:EMC:MATH:TRACe:SHOW

Sets or queries to enable and disable the show recalled trace check box of math trace in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

```
MMEMemory:EMC:MATH:TRACe:SHOW {0 | OFF | 1 | ON} MMEMemory:EMC:MATH:TRACe:SHOW?
```

## Arguments

0 or OFF disable the show recalled trace check box of math trace

1 or ON enable the show recalled trace check box of math trace

## Returns

<Boolean> 0,1

## Examples

```
MMEMemory:EMC:MATH:TRACe:SHOW might return 0, indicating that show recalled trace check box of math trace is disabled.
```

## MMEMemory:EMC:MATH:TRACe:STORE (No Query Form)

Stores the math trace into the specified file in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

```
MMEMemory:EMC:MATH:TRACe:STORe <file name>
```

## Arguments

<file name> = <String> specifies the file to store the math trace with extension as .emcemi

## Returns

None

## Examples

MMEMemory:EMC:MATH:TRACE:STORE "C:\SignalVu-PC Files\math.emcemi" stores the math trace in the math.emcemi file.

## MMEMemory:EMC:RANGe:LOAD:TABLE (No Query Form)

Loads the Ranges table from the specified file in the EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

```
MMEMemory:EMC:RANGe:LOAD:TABLE<file name>
```

## Arguments

<file name> = <String> specifies the file to load the range table with extension as .csv

## Returns

None

## Examples

MMEMemory:EMC:RANGE:LOAD:TABLE "C:\SignalVu-PC Files\table.csv" loads the ranges table from the table.csv file.

## **MMEMemory:EMC:RANGe:STORe:TABLE (No Query Form)**

Stores the Ranges table into the specified file in the EMC view.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMemory:EMC:RANGe:STORe:TABLE <file name>
```

### **Arguments**

<file name> = <String> specifies the file to store the range table with extension as .csv

### **Returns**

None

### **Examples**

MMEMemory:EMC:RANGe:STORe:TABLE "C:\SignalVu-PC Files\sample.csv" stores the ranges table in the sample.csv file.

## **MMEMemory:EMC:RECAll:TRACe<x>:FILE (No Query Form)**

Recalls the specified trace from the file in EMC view.

<x> = <NR1> Either 1,2,3,4,5 (trace number)

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMemory:EMC:RECAll:TRACe<x>:FILE <file name>
```

### **Arguments**

<file name> = <String> specifies the file with extension as .emcemi to recall the trace.

### **Returns**

None

## Examples

`MMEemory:EMC:RECall:TRACe1:FILE "C:\SignalVu-PC Files\sampletrace.emcemi"` recalls the trace 1 from the sampletrace.emcemi file.

## **MMEemory:EMC:RECall:TRACe<x>:SHOW (No Query Form)**

Sets or queries to show or hide the specified recalled trace in EMC view.

<x> = <NR1> Either 1,2,3,4,5 (trace number)

### Conditions

Measurement view: EMC - EMI Display

### Group

Mass Memory commands

### Syntax

```
MMEemory:EMC:RECall:TRACe<x>:SHOW {0 | OFF | 1 | ON}
MMEemory:EMC:RECall:TRACe<x>:SHOW?
```

### Arguments

0 or OFF disables the show check box and hides the recalled trace

1 or ON enables the show check box and displays the recalled trace

### Returns

<Boolean>0,1

## Examples

`MMEemory:EMC:RECall:TRACe2:SHOW` might return 0, indicating recalled trace 2 is hidden.

## **MMEemory:EMC:REPort:GENerate:FINAl (No Query Form)**

Generates the report and saves the report file in the specified format under specified location in EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Mass Memory commands

### Syntax

```
MMEemory:EMC:REPort:GENerate:FINAl <file_name>
```

## Arguments

<file name> = <String> specifies the file and format to store the report.

## Returns

None

## Examples

`MMEMORY:EMC:REPORT:GENERATE:FINAL "C:\SignalVu-PC Files\report.pdf"` generates and stores the report in pdf format with file named report.pdf.

## **MMEMory:EMC:REPOrt:HISTory:DELeTe:ALL (No Query Form)**

Deletes all the saved test results in history list of report setup in EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEMory:EMC:REPOrt:HISTory:DELeTe:ALL`

## Arguments

None

## Returns

None

## Examples

`MMEMORY:EMC:REPORT:HISTORY:DELETE:ALL` deletes all the saved test results in the history list of report setup.

## **MMEMory:EMC:REPOrt:SAVE:RUN (No Query Form)**

Saves the test results and adds to the history list of report setup in EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEMemory:EMC:REPOrt:SAVE:RUN`

## Arguments

None

## Returns

None

## Examples

`MMEMORY:EMC:REPORT:SAVE:RUN` saves the test results and adds to the history list.

## **MMEMemory:EMC:SCAN:TRACe<x>:STORE (No Query Form)**

Stores the specified live scan trace into the specified file in the EMC view.

<x> = <NR1> Trace number

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

`MMEMemory:EMC:SCAN:TRACe<x>:STORE <file name>`

## Arguments

<file name> = <String> specifies the file to store the scan trace with extension as .emcemi

## Returns

None

## Examples

`MMEMemory:EMC:SCAN:TRACe1:STORE "C:\SignalVu-PC Files\sample.emcemi"` stores the scan trace 1 in the sample.emcemi file.

## **MMEMemory:EMC:SPOT:TRACe<x>:STORE (No Query Form)**

Stores the specified live spot trace into the specified file in the EMC view.

<x> = <NR1> Trace number

## Conditions

Measurement view: EMC - EMI Display



## Group

Mass Memory commands

## Syntax

```
MMEMemory:EMC:SPOT:TRACe<x>:STORE <file name>
```

## Arguments

<file name> = <String> specifies the file to store the spot trace with extension as .emcemi

## Returns

None

## Examples

MMEMemory:EMC:SPOT:TRACe2:STORE "C:\SignalVu-PC Files\trace.emcemi" stores the spot trace 2 in the trace.emcemi file.

## [MMEMemory:{FSETtling|PSETtling}:LOAD:TRACe<x>

Loads the Frequency or Phase Settling Time trace data from the specified file. The parameter <x> = 1 or 2; only Trace1 and Trace2 are used for settling time measurements.

## Conditions

Measurement views: Frequency or Phase Settling

## Group

Mass Memory commands

## Syntax

```
[MMEMemory:{FSETtling|PSETtling}:LOAD:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file from which to load trace data.

## Examples

MMEMemory:PSETTLING:LOAD:TRACE1 "Wfm1" loads Trace1 data from the Wfm1 file.

## [MMEMemory:{FSETtling|PSETtling}:SHOW:TRACe<x>

Enables display of a recalled Frequency or Phase Settling trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Trace control panel.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Mass Memory commands

## Syntax

```
[MMEMory:{FSETtling|PSETtling}:SHOW:TRACe<x> { OFF | ON | 0 | 1 }
```

```
[MMEMory:{FSETtling|PSETtling}:SHOW:TRACe<x>?
```

## Related Commands

[\[MMEMory:{FSETtling|PSETtling}:LOAD:TRACe<x>](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

MMEMORY:PSETTLING:SHOW:TRACE1 ON enables display of the recalled data loaded in Trace1.

## [MMEMory:{FSETtling|PSETtling}:STORE:TRACe<x>

Stores the Frequency or Phase Settling trace data in the specified file. The parameter <x> = 1 or 2; only Trace1 and Trace2 are used for Frequency and Phase Settling.

## Conditions

Measurement views: Phase noise

## Group

Mass Memory commands

## Syntax

```
[MMEMory:{FSETtling|PSETtling}:STORE:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data.

## Examples

MMEMORY:PNOISE:STORE:TRACE1 "TRACE1" stores Trace 1 data in the TRACE1 file.

## MMEMory:FVTime[:MEASview<y>]:LOAD:TRACe (No Query Form)

Loads the Frequency versus Time trace data from the specified file.

## Conditions

Measurement views: Frequency versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:FVTime[:MEASview<y>]:LOAD:TRACe <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .FreqVsTime. You can omit the extension.

## Examples

MMEMemory:FVTIME:MEASview1:LOAD:TRACE"Sample1" loads the Frequency versus Time trace data from the *Sample1.FreqVsTime* file.

## MMEMemory:FVTIME[:MEASview<y>]:SHOW:TRACe<x>

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: Frequency versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:FVTIME[:MEASview<y>]:SHOW:TRACe<x> { OFF | ON | 0 | 1 }
MMEMemory:FVTIME:SHOW:TRACe<x>?
```

## Related Commands

[MMEMemory:FVTime\[:MEASview<y>\]:LOAD:TRACe](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

MMEMemory:FVTIME:MEASview1:SHOW:TRACE2 ON enables display of the recalled data loaded in Trace2.

## MMEMemory:FVTime[:MEASview<y>]:STORE:TRACe (No Query Form)

Stores the Frequency versus Time trace data in the specified file.

## Conditions

Measurement views: Frequency versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:FVTime[:MEASview<y>]:STORE:TRACe <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .FreqVsTime. You can omit the extension.

## Examples

MMEMemory:FVTIME:MEASview1:STORE:TRACE"Sample1" stores the Frequency versus Time trace data in the *Sample1.FreqVsTime* file.

## MMEMemory:IQVTime[:MEASVIEW<y>]:LOAD:TRACe:I (No Query Form)

Loads I trace data from the specified file.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:IQVTime[:MEASVIEW<y>]:LOAD:TRACe:I <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .RFIQVsTime. You can omit the extension.

## Examples

MMEMemory:IQVTIME:MEASVIEW1:LOAD:TRACE:I"Sample1" loads the I trace data from the *Sample1.RFIQVsTime* file.

## MMEMemory:IQVTime[:MEASVIEW<y>]:LOAD:TRACe:Q (No Query Form)

Loads Q trace data from the specified file.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:IQVTime[:MEASVIEW<y>]:LOAD:TRACe:Q <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .RFIQVsTime. You can omit the extension.

## Examples

MMEMemory:IQVTIME:MEASVIEW1:LOAD:TRACE:Q"Sample2" loads the I trace data from the *Sample2.RFIQVsTime* file.

## MMEMemory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe:I

Enables display of a recalled trace file in Trace. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe:I { OFF | ON | 0 | 1 }
MMEMemory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe:I?
```

## Related Commands

[MMEMemory:IQVTime\[:MEASVIEW<y>\]:LOAD:TRACe:I](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

`MMEMORY:IQVTIME:MEASVIEW1:SHOW:TRACE:I ON` enables display of the recalled data loaded in the Trace.

## **MMEMory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe<x>:Q**

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting “Show recalled trace” in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe<x>:Q { OFF | ON | 0 | 1 }  
MMEMory:IQVTIME[:MEASVIEW<y>]:SHOW:TRACe<x>:Q?
```

## Related Commands

[MMEMory:IQVTIME\[:MEASVIEW<y>\]:SHOW:TRACe<x>:Q](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

`MMEMORY:IQVTIME:MEASVIEW1:SHOW:TRACE3:Q ON` enables display of the recalled data loaded in Trace3.

## **MMEMory:IQVTime[:MEASVIEW<y>]:STORE:TRACe:I (No Query Form)**

Stores I trace data in the specified file.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMory:IQVTime[:MEASVIEW<y>]:STORE:TRACe:I <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .RFIQVsTime. You can omit the extension.

## Examples

MMEMORY:IQVTIME:MEASVIEW1:STORE:TRACE:I"Sample1" stores the I trace data in the *Sample1.RFIQVsTime* file.

## MMEMory:IQVTime[:MEASVIEW<y>]:STORE:TRACe:Q (No Query Form)

Stores Q trace data in the specified file.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMory:IQVTime[:MEASVIEW<y>]:STORE:TRACe:Q <file_name>
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .RFIQVsTime. You can omit the extension.

## Examples

MMEMORY:IQVTIME:MEASVIEW1:STORE:TRACE:Q"Sample2" stores the Q trace data in the *Sample2.RFIQVsTime* file.

## MMEMory:LOAD:IQ (No Query Form)

Loads time domain IQ waveform into the acquisition memory from a file.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

```
MMEMory:LOAD:IQ <file_name>
```

## Arguments

`<file_name>::<string>` specifies the file to load IQ data from. The file extension `.tiq` is for non-multichannel `tiq` files and `.tiqm` for multichannel `tiq` files. You can omit the extension.

## Examples

`MMEMemory:LOAD:IQ"IQ1"` loads IQ data from the `IQ1.tiq` file.

## MMEMemory:LOAD:STATe (No Query Form)

Loads the instrument setup from a specified file for the currently selected view.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

`MMEMemory:LOAD:STATe <file_name>`

## Arguments

`<file_name>::<string>` specifies the file to load the instrument setup from. The file extension is `.setup`. You can omit the extension.

## Examples

`MMEMemory:LOAD:STATE"STATE1"` loads the instrument setup from the `STATE1.setup` file.

## MMEMemory:LOAD:TRACe (No Query Form)

Loads the trace data from a specified file for the currently selected view.

## Conditions

Measurement views (See Table 41.)

## Group

Mass Memory commands

## Syntax

`MMEMemory:LOAD:TRACe <file_name>`

## Related Commands

[MMEMemory:STORe:TRACe](#)



## Arguments

`<file_name>::<string>` specifies the file to load the trace data from. The file extension is named for the measurement view as shown in the following table. The trace file is not available in some views which are indicated by "NA" in the file extension column.

For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column in the following table. You can load the specified trace with any trace data that you saved in the view if the traces have the same units. For example, you can load Trace 2 with Trace 1 data that you saved.

**Table 42: Trace file extension and trace selection command**

Display group	Measurement view	File extension	Trace selection
General signal viewing	Spectrum	Specan	TRACe<x>:SPECtrum:SElect
	DPX spectrum	dpt	TRACe<x>:DPX:SElect
	Amplitude versus Time	AmplVsTime	TRACe<x>:AVTime:SElect
	Frequency versus Time	FreqVsTime	NA
	Phase versus Time	PhaseVsTime	NA
	RF I&Q versus Time	RFIQVsTime	NA
	Spectrogram	Sogram	NA
	Time overview	NA	NA
General purpose analog modulation (Option 21 only)	AM	AmVsTime	NA
	FM	FmVsTime	NA
	PM	PmVsTime	NA
General purpose digital modulation (Option 21 only)	Constellation	NA	NA
	Demodulated I & Q versus Time	NA	NA
	Eye Diagram	NA	NA
	EVM versus Time	NA	NA
	Frequency Deviation versus Time	NA	NA
	Magnitude error versus Time	NA	NA
	Phase error versus Time	NA	NA
	Signal quality	NA	NA
	Symbol table	NA	NA
	Trellis Diagram	NA	NA
RF measurements	CCDF	CCDF	TRACe<x>:CCDF:SElect
	Channel power and ACPR	NA	NA
	MCPR	NA	NA
	Occupied Bandwidth	NA	NA
	Phase noise (Option 11 only)	PhaseNoise	TRACe<x>: {FSETtling PSETtling}:SElect
	Spurious	NA	NA

Table continued...

Display group	Measurement view	File extension	Trace selection
Pulsed RF	Pulse statistics	NA	NA
	Pulse table	NA	NA
	Pulse trace	NA	NA

## Examples

`MMEMORY:LOAD:TRACE"TRACE1"` loads the trace data from the `TRACE1.Specan` file when the spectrum view is selected.

## MMEMory:LTE:ACLR:LOAD:TABLE (No Query Form)

Loads the LTE ACLR table containing the limits for enabled ranges from the specified `.csv` file.

### Conditions

Measurement view: LTE ACLR

### Group

Mass Memory commands

### Syntax

`MMEMory:LTE:ACLR:LOAD:TABLE <string>`

### Arguments

`<file_name>::<string>` specifies the file from which to load the LTE ACLR table. The file extension is `.csv`. You can omit the extension.

### Examples

`MMEMORY:LTE:ACLR:LOAD:TABLE "Table1"` loads the LTE ACLR table from the `Table1.csv` file.

## MMEMory:LTE:ACLR:STORE:TABLE (No Query Form)

Stores the LTE ACLR table containing the limits for enabled ranges in a specified `.csv` file, allowing you to export the file into Microsoft Excel or other database system.

### Conditions

Measurement view: LTE ACLR

### Group

Mass Memory commands

### Syntax

`MMEMory:LTE:ACLR:STORE:TABLE <string>`

## Arguments

`<file_name>::<string>` specifies the file in which to store the LTE ACLR table.

The file extension is `.csv`. You can omit the extension.

## Examples

`MMEMORY:LTE:ACLR:STORE:TABLE "Table1"` stores the LTE ACLR table in the `Table1.csv` file.

## MMEMory:MAPit:LOAD (No Query Form)

Loads the MapIt measurement results from the specified file.

## Conditions

Measurement views: Spectrum, DPX, Spectrogram, Amplitude vs Time, Channel Power, Occupied Bandwidth

## Group

Mass Memory commands

## Syntax

`MMEMory:MAPit:LOAD <filename>`

## Arguments

`<filename>::<string>` the name of the file from which to load the MapIt data.

## Examples

`MMEMORY:MAPIT:LOAD "C:\SignalVu-PC files\MapIt_results1"` Loads the MapIt measurement results in the `MapIt_results1` file from the `C:SignalVu-PC files` folder.

## MMEMory:MAPit:STORe (No Query Form)

Save the selected MapIt measurement results to a file. The results to save are selected by the `<result_index>`. The saved files are compressed `.zip` files.

## Conditions

Measurement views: Spectrum, DPX, Spectrogram, Amplitude vs Time, Channel Power, Occupied Bandwidth

## Group

Mass Memory commands

## Syntax

`MMEMory:MAPit:STORe <filename>, <result_index> [,<result_index>]`

## Arguments

`<filenam>::<string>` the name of the file in which to store the MapIt measurement results.

<result\_index>::<Nrf>. The 1-based index of the MapIt result to load. See the MapIt editor for the list of results. index 1 is the first result at the top of the list.

## Examples

```
MMEMORY:MAPIT:STORE ??
```

## **MMEMory:MAPit:STORe:ALL (No Query Form)**

Saves all of the MapIt measurement results from memory to the specified file.

### Conditions

Measurement views: Spectrum, DPX, Spectrogram, Amplitude vs Time, Channel Power, Occupied Bandwidth

### Group

Mass Memory commands

### Syntax

```
MMEMory:MAPit:STORe:ALL <filename>
```

### Arguments

<filename>::<string> the name of the file in which to store the MapIt measurement results.

## Examples

```
MMEMORY:MAPIT:STORE:ALL "C:\SignalVu-PC files\MapIt_results1" Stores the MapIt measurement results in the MapIt_results1 file in the C:SignalVu-PC files folder.
```

## **MMEMory:NOISe:GAIN:LOAD:TRACe (No Query Form)**

Loads the Noise Gain trace data from the specified file.

### Conditions

Measurement view: Gain

### Group

Mass Memory commands

### Syntax

```
MMEMory:NOISe:GAIN:LOAD:TRACe <file_name>
```

### Related Commands

[MMEMory:NOISe:GAIN:STORe:TRACe](#)

### Arguments

<file\_name> = the file from which to load trace data.

## Examples

`MMEMORY:NOISE:GAIN:LOAD:TRACe`trace1 will load the trace1 data file.

## MMEMory:NOISe:GAIN:SHOW:TRACe

Show or hide the Noise Gain trace from the specified file.

### Conditions

Measurement view: Gain display

### Group

Mass Memory commands

### Syntax

```
MMEMory:NOISe:GAIN:SHOW:TRACe { OFF | ON | 0 | 1 } MMEMory:NOISe:GAIN:SHOW:TRACe
```

### Arguments

OFF or 0 sets the trace to hide from the specified file.

ON or 1 sets the trace to show from the specified file.

### Returns

0 means the trace is hidden.

1 means the trace is showing.

## Examples

`MMEMORY:NOISE:GAIN:SHOW:TRACE?` might return 1, indicating that the trace is set to show from the specified file.

## MMEMory:NOISe:GAIN:STORe:TRACe (No Query Form)

Stores the Noise Gain trace data in the specified file.

### Conditions

Measurement view: Gain display

### Group

Mass Memory commands

### Syntax

```
MMEMory:NOISe:GAIN:STORe:TRACe <file_name>
```

### Related Commands

[MMEMory:NOISe:GAIN:LOAD:TRACe](#)

## Arguments

<file\_name> = the file in which to store trace data.

## Examples

```
MMEMORY:NOISE:GAIN:STORE:TRACE
```

## MMEMory:Noise:LOAD:ENRCALibration (No Query Form)

Loads the specified Noise Figure ENR table.

## Conditions

Measurement view: Noise display

## Group

Mass Memory commands

## Syntax

```
MMEMory:Noise:LOAD:ENRCALibration <filename>
```

## Syntax

```
MMEMory:Noise:LOAD:ENRCALibration <filename>
```

## Arguments

<file\_name> = the file from which to load ENR calibration data.

## Examples

```
MMEMORY:NOISE:LOAD:ENRCALIBRATION
```

## MMEMory:Noise:LOAD:FREQTABLE (No Query Form)

Loads the specified Noise Figure Frequency table.

## Conditions

Measurement view: Noise display

## Group

Mass Memory commands

## Syntax

```
MMEMory:Noise:LOAD:FREQTABLE <filename>
```

## Arguments

<filename> = the file from which to load the frequency table data.

---

## Examples

```
MMEMORY:NOISE:LOAD:FREQTABLE
```

## **MMEMory:Noise:STORe:ENRCALibration (No Query Form)**

Saves the Noise Figure ENR table to the specified location.

### Conditions

Measurement view: Noise display

### Group

Mass Memory commands

### Syntax

```
MMEMory:Noise:STORe:ENRCALibration
```

### Arguments

<filename> = the file in which to store ENR calibration data.

## Examples

```
MMEMORY:NOISE:STORE:ENRCALIBRATION
```

## **MMEMory:Noise:STORe:FREQTABLE (No Query Form)**

Saves the Noise Figure Frequency table to the specified location.

### Conditions

Measurement view: Noise display

### Group

Mass Memory commands

### Syntax

```
MMEMory:Noise:STORe:FREQTABLE <filename>
```

### Arguments

<filename> = the file in which to store the frequency table data.

## Examples

```
MMEMORY:NOISE:STORE:FREQTABLE
```

## **MMEMory:OFDM:CONSte:LOAD:TRACe (No Query Form)**

Loads the OFDM constellation trace data from the specified file.

### **Conditions**

Measurement views: OFDM

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:OFDM:CONSte:LOAD:TRACe <file_name>
```

### **Arguments**

<file\_name>::<string> specifies the file from which to load the trace data.

### **Examples**

MMEMORY:OFDM:CONSTE:LOAD:TRACE "Sample1" loads the trace data from the file with the OFDM Constellation measurement.

## **MMEMory:OFDM:CONSte:SHOW:TRACe**

Enables the display of a recalled trace.

### **Conditions**

Measurement views: OFDM

### **Group**

Mass Memory commands

### **Syntax**

```
MMEMory:OFDM:CONSte:SHOW:TRACe{ OFF | ON | 0 | 1 } MMEMory:OFDM:CONSte:SHOW:TRACe?
```

### **Arguments**

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

### **Examples**

MMEMORY:OFDM:CONSTE:SHOW:TRACE ON enables the display of the recalled data loaded in the trace.

## **MMEMory:OFDM:CONSte:STORe:TRACe (No Query Form)**

Stores the OFDM constellation measurement trace data in the specified file.



## Conditions

Measurement views: OFDM

## Group

Mass Memory commands

## Syntax

```
MMEMemory:OFDM:CONStE:StORe:TRACe <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store the trace data.

## Examples

`MMEMemory:OFDM:CONStE:StORe:TRACe "Sample1"` stores the trace data in the Sample1 file in the OFDM Constellation measurement.

## MMEMemory:PHVTime[:MEASVIEW<y>]:LOAD:TRACe (No Query Form)

Loads the Phase versus Time trace data from the specified file.

## Conditions

Measurement views: Phase versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PHVTime[:MEASVIEW<y>]:LOAD:TRACe <file_name>
```

## Arguments

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .PhaseVsTime. You can omit the extension.

## Examples

`MMEMemory:PHVTime:MEASVIEW1:LOAD:TRACe"Sample1"` loads the Phase versus Time trace data from the *Sample1.PhaseVsTime* file.

## MMEMemory:PHVTime[:MEASview<y>]:SHOW:TRACe

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: Phase versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PHVTime[:MEASview<y>]:SHOW:TRACe { OFF | ON | 0 | 1 }  
MMEMemory:PHVTime:SHOW:TRACe?
```

## Related Commands

[MMEMemory:PHVTime\[:MEASview<y>\]:LOAD:TRACe](#)

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

MMEMemory:PHVTIME:MEASview1:SHOW:TRACe ON enables display of the recalled data loaded in theTrace.

## MMEMemory:PHVTime[:MEASVIEW<y>]:STORE:TRACe (No Query Form)

Stores the Phase versus Time trace data in the specified file.

## Conditions

Measurement views: Phase versus Time

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PHVTime[:MEASVIEW<y>]:STORE:TRACe <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .PhaseVsTime. You can omit the extension.

## Examples

MMEMemory:PHVTIME:MEASVIEW1:STORE:TRACe"Sample1" stores the Phase versus Time trace data in the *Sample1.PhaseVsTime* file.

## MMEMemory:PNOise:LOAD:TRACe<x> (No Query Form)

Loads the Phase Noise trace data from the specified file. The parameter <x> = 1 or 2; only Trace1 and Trace2 are used for Phase Noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PNOise:LOAD:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file from which to load trace data.

## Examples

```
MMEMORY:PNOISE:LOAD:TRACE1 "Wfm1" loads Trace1 data from the Wfm1 file.
```

## MMEMemory:PNOise:SHOW:TRACe<x>

Enables display of a recalled Phase Noise trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: Phase noise

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PNOise:SHOW:TRACe<x> { OFF | ON | 0 | 1 } MMEMemory:PNOise:SHOW:TRACe<x>?
```

## Related Commands

[MMEMemory:PNOise:LOAD:TRACe<x>](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

```
MMEMORY:PNOISE:SHOW:TRACE1 ON enables display of the recalled data loaded in Trace1.
```

## MMEMemory:PNOise:STORe:TRACe<x> (No Query Form)

Stores the Phase Noise trace data in the specified file. The parameter <x> = 1 or 2; only Trace1 and Trace2 are used for Phase Noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Mass Memory commands

## Syntax

```
MMEMemory:PNOise:STORe:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data.

## Examples

```
MMEMORY:PNOISE:STORE:TRACE1 "WFM1" stores Trace 1 data in the WFM1 file.
```

## MMEMemory:SEM:LOAD:TABLE (No Query Form)

Loads the Spectral Emissions Mask table containing the limits for enabled ranges from the specified file.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SEM:LOAD:TABLE <file_name>
```

## Arguments

<file\_name>::<string> specifies the file to load the SEM table from. The file extension is `.csv`. You can omit the extension.

## Examples

```
MMEMORY:SEM:LOAD:TABLE "Table1" loads the SEM table from the Table1.csv file.
```

## MMEMemory:SEM:STORe:TABLE (No Query Form)

Stores the Spectral Emissions Mask table containing the limits for enabled ranges in a specified file in the CSV format, allowing you to export the file into Microsoft Excel or other database systems.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SEM:STORe:TABLE <file_name>
```

## Arguments

<file\_name>::<string> specifies the file to store the SEM table in. The file extension is .csv. You can omit the extension.

## Examples

```
MMEMemory:SEM:STORe:TABLE "Table1" stores the SEM table in the Table1.csv file.
```

## MMEMemory:SGRam[:MEASview<y>]:LOAD:TRACe (No Query Form)

Loads the Spectrogram trace data from the specified file. No trace number is needed or allowed.

## Conditions

Measurement views: Spectrogram

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SGRam[:MEASview<y>]:LOAD:TRACe <file_name>
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .sogram. You can omit the extension.

## Examples

```
MMEMemory:SGRam:MEASview1:LOAD:TRACe5 "Spec2" loads Trace5 data from the file
```

*Spec2.sogram.*

## MMEMemory:SGRam[:MEASview<y>]:SHOW:TRACe

Enables display of a recalled trace file in Trace. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: Spectrogram

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SGRam[:MEASview<y>]:SHOW:TRACe { OFF | ON | 0 | 1 }
```

```
MMEMemory:SGRam[:MEASview<y>]:SHOW:TRACe?
```

## Related Commands

```
MMEMemory:SGRam[:MEASview<y>]:LOAD:TRACe
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Returns

```
{ OFF | ON | 0 | 1 }
```

## Examples

```
MMEMemory:SGRam:MEASview1:SHOW:TRACe ON enables display of the recalled data loaded in Trace.
```

## MMEMemory:SGRam[:MEASview<y>]:STORE:TRACe (No Query Form)

Stores the Spectrogram trace data in the specified file. No trace number is needed or allowed.



**Note:** Only valid when the spectrum or spectrogram measurements are running.

## Conditions

Measurement views: Spectrogram

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SGRam[:MEASview<y>]:STORE:TRACe <file_name>
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .sogram. You can omit the extension.

## Examples

`MMEMory:SGRam:MEASview1:STORe:TRACe "Spec2"` stores Trace data in the *Spec2.sogram* file.

## MMEMory:SGRam[:MEASview<y>]:STORe:PLOT

Stores the spectrogram plot data into a specified location and file.

### Conditions

Measurement views: Spectrogram

### Group

Mass Memory commands

### Syntax

`MMEMory:SGRam[:MEASview<y>]:STORe:PLOT <file_name>`

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<file\_name> = <String> specifies the file location and file name to which the plot to be saved.

### Returns

None

## Examples

`MMEMory:SGRam:MEASview1:STORe:PLOT "C:\SignalVu-PC Files\Sample1.Specan"` saves the plot in Sample1.Specan file.

## MMEMory:SGRam[:MEASview<y>]:STORe:PLOT:DONe (Query only)

Determines whether the spectrogram plot data is saved or not saved.

### Conditions

Measurement views: Spectrogram

### Group

Mass Memory commands

### Syntax

`MMEMory:SGRam[:MEASview<y>]:STORe:PLOT:DONe?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<Boolean> 0, 1

1 indicates the spectrogram plot data is saved.

0 indicates the spectrogram plot data is not saved.

## Examples

MMEMory:SGRam:MEASview1:STORe:PLOT:DONe? might return 1, indicating the spectrogram plot data is saved.

## MMEMory:SPECTrum:MEASview<y>:LOAD:TRACe (No Query Form)

Loads the spectrum trace data from the specified file.

The parameter <x> = 1 to 4; Trace 5 (spectrogram) is not valid.

## Conditions

Measurement views: Spectrum

## Group

Mass Memory commands

## Syntax

```
MMEMory:SPECTrum:MEASview<y>:LOAD:TRACe <file_name>
```

## Arguments

<file\_name>::<string> specifies the file from which to load trace data. The file extension is .Specan. You can omit the extension.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

MMEMory:SPECTrum:MEASview2:LOAD:TRACe "C:\SignalVu-PC Files\Sample1.Specan" loads Trace 1 data from Sample1.Specan file

## MMEMory:SPECTrum[:MEASview<y>]:SHOW:TRACe<x>

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

## Conditions

Measurement views: Spectrum



## Group

Mass Memory commands

## Syntax

```
MMEMemory:SPECTrum[:MEASview<y>]:SHOW:TRACe<x> { OFF | ON | 0 | 1 }
```

```
MMEMemory:SPECTrum[:MEASview<y>]:SHOW:TRACe<x>?
```

## Related Commands

[MMEMemory:SPECTrum\[:MEASview<y>\]:LOAD:TRACe](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
MMEMORY:SPECTRUM:MEASVIEW2:SHOW:TRACE2 ON enables display of the recalled data loaded in Trace2
```

## MMEMemory:SPECTrum[:MEASview<y>]:STORE:PLOT (No Query Form)

Stores the spectrum plot data into a specified location and file.

## Conditions

Measurement view: Spectrum

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SPECTrum[:MEASview<y>]:STORE:PLOT <file_name>
```

## Arguments

<file\_name> = <String> specifies the file location and file name to which the plot to be saved.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

None

## Examples

MMEMORY:SPECTRUM:MEASVIEW2:STORE:PLOT"C:\SignalVu-PC Files\Sample1.Specan" saves the plot in Sample1.Specan file

## MMEMory:SPECTrum[:MEASview<y>]:STORE:PLOT:DONe (Query Only)

Determines whether the spectrum plot data is saved or not saved.

### Conditions

Measurement view: Spectrum

### Group

Mass Memory commands

### Syntax

MMEMory:SPECTrum[:MEASview<y>]:STORE:PLOT:DONe?

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

<Boolean> 0, 1

1 indicates the spectrum plot data is saved

0 indicates the spectrum plot data is not saved

## Examples

MMEMORY:SPECTRUM:MEASVIEW2:STORE:PLOT:DONE might return 1, indicating the spectrum plot data is saved

## MMEMory:SPECTrum[:MEASview<y>]:STORE:TRACe(No Query Form)

Stores the spectrum trace data in the specified file. The parameter <x> = 1 to 5; All traces are valid.



**Note:** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

### Conditions

Measurement views: Spectrum

### Group

Mass Memory commands

### Syntax

MMEMory:SPECTrum[:MEASview<y>]:STORE:TRACe<file\_name>

## Arguments

`<file_name>::<string>` specifies the file in which to store trace data. The file extension is .Specan. You can omit the extension.

The file extension is named for the measurement view as shown in the table (Click here for Trace file extension and trace selection command) The trace file is not available in some views which are indicated by "NA" in the file extension column. For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column.

The parameter `<y>=1` to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`MMEMORY:SPECTRUM:MEASview2:STORE:TRACE"TRACE1"` stores the trace data in the TRACE1.Specan file when the spectrum view is selected.

## MMEMory:SPURious:LOAD:TABLE (No Query Form)

Loads the spurious table containing the limits for enabled ranges from the specified file.

### Conditions

Measurement views: Spurious

### Group

Mass Memory commands

### Syntax

`MMEMory:SPURious:LOAD:TABLE <file_name>`

### Arguments

`<file_name>::<string>` specifies the file to load the spurious table from. The file extension is .csv. You can omit the extension.

### Examples

`MMEMORY:SPURIOUS:LOAD:TRACE1 "Table1"` loads the spurious table from the *Table1.csv* file.

## MMEMory:SPURious:RECALL:TRACe (No Query Form)

Recalls the spurious trace data from the specified file and trace number.

### Conditions

Measurement view: Spurious Display

### Group

Mass Memory commands

### Syntax

`MMEMory:SPURious:RECALL:TRACe <x> <file_name>`

## Arguments

<x> = 1,2,3,4 is the trace <file\_name> = <String> specifies the file location and file name to which the trace data to be Recalled

## Returns

None

## Examples

```
MMEMORY:SPURIOUS:RECALL:TRACE "C:\SignalVu-PC Files\Sample1.spurious"
```

recalls the Trace 3 data from the Sample1.spurious file.

## MMEMory:SPURious:SHOW:TRACe (No Query Form)

Displays the spurious trace data from the specified file and trace number.

## Conditions

Measurement view: Spurious Display

## Group

Mass Memory commands

## Syntax

```
MMEMory:SPURious:SHOW:TRACe <x> <file_name>
```

## Arguments

<x> = 1,2,3,4 is the trace <file\_name> = <String> specifies the file location and file name to which the trace data to be displayed.

## Returns

None

## Examples

```
MMEMORY:SPURIOUS:SHOW:TRACE "C:\SignalVu-PC Files\Sample1.spurious" displays the Trace 2 data from Sample1.spurious file.
```

## MMEMory:SPURious:STORe:TABLE (No Query Form)

Stores the spurious table containing the limits for enabled ranges in a specified file in the CSV (Comma Separated Values) format, allowing you to export the file into Microsoft Excel or other database systems.

## Conditions

Measurement views: Spurious

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SPURious:STORe:TABLE <file_name>
```

## Arguments

<file\_name>::<string> specifies the file to store the spurious table in. The file extension is .csv. You can omit the extension.

## Examples

```
MMEMemory:SPURIOUS:STORE:TABLE "Table1" stores the spurious table in the Table1.csv file.
```

## MMEMemory:SPURious:STORe:TRACe (No Query Form)

Stores the specified spurious trace data into a specified location and file.

## Conditions

Measurement view: Spurious Display

## Group

Mass Memory commands

## Syntax

```
MMEMemory:SPURious:STORe:TRACe <x> <file_name>
```

## Arguments

<x> = 1,2,3,4 is the trace <file\_name> = <String> specifies the file location and file name to which the trace data to be saved.

## Returns

None

## Examples

```
MMEMemory:SPURIOUS:STORE:TRACE "C:\SignalVu-PC Files\Sample1.spurious" saves the Trace 1 data in Sample1.spurious file.
```

## MMEMemory:STORe:IQ (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ <file_name>
```

## Arguments

<file\_name>::

The file extension .tiq is for non-multichannel setup and .tiqm for multichannel setup. You can omit the extension.

## Examples

```
MMEMORY:STORE:IQ "IQ1" saves IQ data to the IQ1.tiq file.
```

## MMEMemory:STORe:IQ:CSV (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file in the CSV (Comma Separated Values) format, allowing you to export the file into Microsoft Excel or other database systems.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ:CSV <file_name>
```

## Arguments

<file\_name>::

## Examples

```
MMEMORY:STORE:IQ:CSV "IQ2" saves IQ data to the IQ2.cs(less t eq) file.
```

## MMEMemory:STORe:IQ:MAT (No Query Form)

Saves time-domain IQ waveform in the acquisition memory to a specified file in the MATLAB format, allowing you to export the file into the MATLAB technical computing environment.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ:MAT <file_name>
```

## Arguments

<file\_name>::<string> specifies the file to save IQ data. The file extension is .mat. You can omit the extension.

## Examples

```
MMEMemory:STORE:IQ:MAT "IQ3" saves IQ data to the IQ3.mat file.
```

## MMEMemory:STORe:IQ:SELEct:DATA

Sets or queries the data type stored when saving acquisition data files.

## Conditions

Measurement views

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ:SELEct:DATA { IQ | SPECTra | ALL }
```



**Note:** Although the command includes IQ as part of the command, the command applies to all the data stored in the file, not only IQ data.

## Arguments

**IQ** specifies that IQ data be saved in the data file.

**SPECTra** specifies that DPX spectra be saved in the data file.

**ALL** specifies that both IQ and DPX spectra be saved in the data file.

## Examples

```
MMEMemory:STORE:IQ:SELECT:DATA SPECTRA sets the data type to DPX Spectra.
```

## MMEMemory:STORe:IQ:SELEct:FRAMES

Sets or queries which frames are saved when saving TIQ acquisition data files.

## Conditions

Measurement views

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ:SELEct:FRAMes { ACQquisition | HISTory | SELEcted | ALL }
```

## Arguments

ACQquisition saves the entire acquisition that contains the current frame.

HISTory saves the current frame. SELEcted saves only selected frames. ALL saves all acquired frames.

## MMEMemory:STORe:IQ:SELEct:LENGth

Sets or queries what portion of each record is saved when saving acquisition data.

## Conditions

Measurement views

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:IQ:SELEct:LENGth { ENTire | LENGth }
```

```
MMEMemory:STORe:IQ:SELEct:LENGth?
```

## Arguments

ENTire saves the entire record

LENGth saves the portion of the record being analyzed.

## MMEMemory:STORe:MSTate (No Query Form)

Stores the measurement parameters to a specified file in the ASCII text format for the currently selected view, allowing you to export the file into other applications.

## Conditions

Measurement views: All

## Group

Mass Memory commands

## Syntax

```
MMEMemory:STORe:MSTate <file_name>
```

## Arguments

<file\_name>::<string> specifies the file to store the measurement parameters. The file extension is .txt. You can omit the extension.



---

## Examples

`MMEMORY:STORE:MSTATE "MSTATE1"` stores the measurement parameters to the *MSTATE1.txt* file.

## MMEMory:STORe:RESults (No Query Form)

Stores the measurement results including measurement parameters and trace data to a specified file in the CSV (Comma Separated Values) format for the currently selected view, allowing you to export the file into Microsoft Excel or other database systems.

### Conditions

Measurement views: All

### Group

Mass Memory commands

### Syntax

```
MMEMory:STORe:RESults <file_name>
```

### Arguments

`<file_name>::<string>` specifies the file to store the measurement results.

The file extension is `.csv`. You can omit the extension.

## Examples

`MMEMORY:STORE:RESULTS "RESULT1"` stores the measurement results to the *RESULT1.csv* file.

## MMEMory:STORe:SCReen (No Query Form)

Stores the current display as a bitmap image file in one of several standard formats. When no format is specified, the system uses the default, `.png`.

### Conditions

Measurement views: All

### Group

Mass Memory commands

### Syntax

```
MMEMory:STORe:SCReen <file_name>{ .bmp | .jpg | .png }
```

### Arguments

`<file_name>::<string>` specifies the file to store the screen image.

## Examples

`MMEMORY:STORE:SCREEN "RESULT1.png"` stores the current screen image into the file `RESULT1.png`.

## MMEMory:STORe:STATe (No Query Form)

Stores the instrument setup to a specified file for the currently selected view.

### Conditions

Measurement views: All

### Group

Mass Memory commands

### Syntax

`MMEMory:STORe:STATe <file_name>`

### Arguments

`<file_name>::<string>` specifies the file to store the instrument setup. The file extension is setup. You can omit the extension.

### Examples

`MMEMORY:STORE:STATE "STATE1"` stores the instrument setup in the `STATE1.setup` file.

## MMEMory:STORe:TRACe (No Query Form)

Stores the trace data in a specified file for the currently selected view.

### Conditions

Measurement views (See Table 41 on page 2-1357.)

### Group

Mass Memory commands

### Syntax

`MMEMory:STORe:TRACe <file_name>`

### Related Commands

[MMEMory:IQVTIME:SHOW:TRACe:l](#)

### Arguments

`<file_name>::<string>` specifies the file to store the trace data in.

The file extension is named for the measurement view as shown in the table (See Table 2-28.) The trace file is not available in some views which are indicated by "NA" in the file extension column. For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column.

## Examples

`MMEMORY:STORE:TRACE "TRACE1"` stores the trace data in the *TRACE1.Specan* file when the spectrum view is selected.

## **MMEMory:TOVerview[:MEASview<y>]:LOAD:TRACe1 (No Query Form)**

Loads the trace data from a specified file into Trace1. Refer to (See Table 2-28.) for the correct file extensions for each trace type.

### Conditions

Measurement views

### Group

Mass Memory commands

### Syntax

`MMEMory:TOVerview[:MEASview<y>]:LOAD:TRACe1 <file_name>`

### Related Commands

[MMEMory\[:MEASview<y>\]:STORe:TRACe](#)

### Arguments

`<file_name>::<string>` specifies the file path and name to load the trace data from. The file extension is named for the measurement view.(See Table 2-28.).

Most trace file types are compatible with only one Measurement view.

### Examples

`MMEMORY:TOVERVIEW:MEASVIEW1:LOAD:TRACE1 "TrialB"`  
loads the trace data from the file *TRIALB.TOV* when the spectrum view is selected.

## **MMEMory:TOVerview[:MEASview<y>]:SHOW:TRACe1**

Enables display of a recalled trace file in Trace<x>. The result is the same as selecting "Show recalled trace" in the Recall traces dialog box or selecting Show in the Trace Overview display.

### Conditions

Measurement views

### Group

Mass Memory commands

## Syntax

```
MMEMemory:TOVerview[:MEASview<y>:SHOW]:TRACe1 { OFF | ON | 0 | 1 }  
MMEMemory:TOVerview[:MEASview<y>]:SHOW:TRACe1?
```

## Related Commands

[MMEMemory:TOVerview\[:MEASview<y>\]:LOAD:TRACe1](#)

## Arguments

OFF or 0 disables display of the recalled trace.

ON or 1 enables display of the recalled trace.

## Examples

```
MMEMORY:TOVERVIEW:MEASVIEW1:SHOW:TRACE1 ON enables display of the recalled data loaded in Trace2.
```

## MMEMemory:TOVerview[:MEASview<y>]:STORE:TRACe1 (No Query Form)

Stores the trace data for Trace1 into the specified file.

## Conditions

Measurement views

## Group

Mass Memory commands

## Syntax

```
MMEMemory:TOVerview[:MEASview<y>]:STORE:TRACe1 <file_name>
```

## Related Commands

[MMEMemory\[:MEASview<y>\]:STORE:TRACe](#)

## Arguments

<file\_name>::<string>specifies the file to store the trace data in.

The file extension is named for the measurement view as shown in the table (See Table 2-28.) The trace file is not available in some views which are indicated by "NA" in the file extension column. For the views that allow multiple-trace display, select the trace to load or store using the command listed in the trace selection column.

## Examples

```
MMEMORY:TOVERVIEW:MEASVIEW1:STORE:TRACE1 "TrialB" stores the Trace1 data into the file TRIALB.TOV when a spectrum view is selected.
```

## MMEMemory:TXGain:LOAD:TRACe<x> (No Query Form)

Load the specified saved trace from a file.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace 1, 2, or 3.

## Group

Mass Memory commands

## Syntax

```
MMEMemory:TXGain:LOAD:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file from which to load trace data.

## Examples

MMEMemory:TXGain:LOAD:TRACe1 "sample1" loads the trace data from the sample1.tgt file.

## MMEMemory:TXGain:SHOW:TRACe<x>

Displays a saved trace instead of a live trace.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace 1, 2, or 3.

## Group

Mass Memory commands

## Syntax

```
MMEMemory:TXGain:SHOW:TRACe<x> { OFF | ON | 0 | 1 } MMEMemory:TXGain:SHOW:TRACe<x>?
```

## Arguments

OFF or 0 disables display of the recalled trace. ON or 1 enables display of the recalled trace.

## Returns

OFF or 0 means that display of the recalled trace is disabled. ON or 1 means that display of the recalled trace is enabled.

## Examples

MMEMemory:TXGain:SHOW:TRACe2 ON enables display of the recalled data loaded in Trace 2.

## MMEMemory:TXGain:STORe:TRACe<x> (No Query Form)

Saves the selected trace to a file for later recall and analysis.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace 1, 2, or 3.

## Group

Mass Memory commands

## Syntax

```
MMEMemory:TXGain:STORe:TRACe<x> <file_name>
```

## Arguments

<file\_name>::<string> specifies the file in which to store trace data. The file extension is .tgt. You can omit the extension.

## Examples

```
MMEMemory:TXGain:STORe:TRACe1 "Sample1" stores Trace 1 data in the Sample1.tgt file.
```

## MMEMemory:WLAN:AD:SUMMARY:LIMIT:TABLE:LOAD (No Query Form)

Loads the Summary Limit table containing the limits for enabled ranges from the specified file in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Mass Memory commands

## Syntax

```
MMEMemory:WLAN:AD:SUMMARY:LIMIT:TABLE:LOAD <file name>
```

## Arguments

<file name>= <String> specifies the file to load the Summary Limit table from the file extension as .csv

## Returns

None

## Examples

```
MMEMemory:WLAN:AD:SUMMARY:LIMIT:TABLE:LOAD Table1 loads the Summary Limit table from the Table1 .csv file.
```

## MMEMemory:WLAN:AD:SUMMARY:LIMIT:TABLE:SAVE (No Query Form)

Stores the Summary Limit table containing the limits for enabled ranges in a specified file in the cvs format, allowing you to export the file into Microsoft Excel or other database systems in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Mass Memory commands

## Syntax

```
MMEMemory:WLAN:AD:SUMMary:LIMIt:TABLE:SAVE<file_name>
```

## Arguments

<file\_name>=<string> specifies the file to store the SEM table in the file extension as .csv.

## Returns

None

## Examples

```
MMEMORY:WLAN:AD:SUMMARY:LIMIT:TABLE:SAVE "Table1" stores the SEM table in the Table1.csv file.
```

## MMEMemory:WLAN:AY:SUMMary:LIMIt:TABLE:LOAD (No Query Form)

Loads the Summary Limit Table containing limits for enabled ranges from the specified .csv file.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Mass memory commands

## Syntax

```
MMEMemory:WLAN:AY:SUMMary:LIMIt:TABLE:LOAD <file_name>
```

## Arguments

<file\_name>::=<string> specifies the file to load the Summary Limit Table from a .csv file.



**Note:** You can omit the file extension.

## Examples

```
MMEM:WLAN:AY:SUMM:LIMI:TABL:LOAD "Table1" loads the Summary Limit Table from file Table1.csv.
```

## MMEMemory:WLAN:AY:SUMMary:LIMIt:TABLE:SAVE (No Query Form)

Stores the Summary Limit Table containing limits for enabled ranges in a specified .csv file, allowing you to export the file into Microsoft Excel or other database systems.

This command will save both enabled and disabled entities.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Mass memory commands

## Syntax

```
MMEMory:WLAN:AY:SUMMary:LIMIT:TABLE:SAVE <file_name>
```

## Arguments

<file\_name>::=<string> specifies the .csv file to store the SEM table.



**Note:** You can omit the file extension.

## Examples

```
MMEM:WLAN:AY:SUMM:LIMI:TABL:SAVE "Table2" stores the SEM table in the file Table2.csv.
```

## MMEMory:WLAN:CRESpOse:LOAD:TRACe<x> (No Query Form)

Loads the selected WLAN Channel Response trace data from the specified file.

<x> is the selected trace, 1 or 2. 1 is the magnitude trace. 2 is the phase trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Mass Memory commands

## Syntax

```
MMEMory:WLAN:CRESpOse:LOAD:TRACe<x> <QString>
```

## Arguments

Quoted string that specifies the file from which to load trace data.

## Examples

```
MMEMORY:WLAN:CRESPONSE:LOAD:TRACE2 "Sample1" loads the WLAN Channel Response trace data from the Sample1 file into Trace 2 (phase trace).
```

## MMEMory:WLAN:CRESpOse:SHOW:TRACe<x>

Enables the display of a recalled WLAN Channel Response trace.

<x> is the selected trace, 1 or 2. 1 is the magnitude trace. 2 is the phase trace.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Mass Memory commands

## Syntax

```
MMEMemory:WLAN:CRESPonse:SHOW:TRACe<x> {0|1|OFF|ON}
```

```
MMEMemory:WLAN:CRESPonse:SHOW:TRACe<x>?
```

## Arguments

ON or 1 turns on the display of a recalled trace.

OFF or 0 turns off the display of a recalled trace.

## Examples

MMEMemory:WLAN:CRESPONSE:SHOW:TRACE2 ON turns on the display of recalled trace 2.

## MMEMemory:WLAN:CRESPonse:STORE:TRACe<x> (No Query Form)

Stores the WLAN Channel Response measurement trace data in the specified file.

<x> is the selected trace, 1 or 2. 1 is the magnitude trace. 2 is the phase trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Mass Memory commands

## Syntax

```
MMEMemory:WLAN:CRESPonse:STORE:TRACe<x> <QString>
```

## Arguments

Quoted string that specifies the file in which to store the trace data.

## Examples

MMEMemory:OFDM:CRESPONSE:STORE:TRACE2 "Sample1" stores the trace data from Trace 2 into the Sample1 file.

## TEST

Resets the trigger system and places all trigger sequences in the idle state. Any actions related to the trigger system that are in progress, such as a sweep or acquiring a measurement is also aborted.

To start data acquisition, use the INITiate commands.

### Conditions

Measurement views: All

### Group

Abort commands

### Syntax

TEST

### Related Commands

INITiate:CONTinuous, INITiate:RESume

### Arguments

None

### Examples

TEST resets the trigger system and stops data acquisition.

## \*OPC

Generates the operation complete message in the Standard Event Status Register (SESR) when all pending operations finish. The \*OPC? query places the ASCII character "1" into the output queue when all pending operations are finished. The \*OPC? response is not available to read until all pending operations finish.

The \*OPC command allows you to synchronize the operation of the analyzer with your application program. Refer to *Synchronizing Execution* (See page 3-11.) for the details.

### Conditions

Measurement views: All

### Group

IEEE common commands

### Syntax

\*OPC

\*OPC?

## Arguments

None

## \*OPT? (Query Only)

Returns a list of options installed in your analyzer.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

\*OPT?

## Arguments

None

## Returns

The numbers of all the options installed in the analyzer, separated by commas. If no options have been installed, 0 is returned. The following table lists the options for the RSA Series analyzers.

**Table 43: Instrument options**

Option	Description
09	Enhanced real time
10	Audio analysis
11	Phase noise and jitter measurement
12	Settling time measurement (frequency and phase)
21	General purpose modulation analysis
22	Flexible OFDM analysis
23	WLAN 802.11a/b/g/j/p analysis
24	WLAN 802.11n analysis
25	WLAN 802.11ac analysis
B25	25 MHz real-time capture bandwidth
B40	40 MHz real-time capture bandwidth
B85	85 MHz real-time capture bandwidth
B16x	165 MHz real-time capture bandwidth
50	1 MHz to 3/6.2 GHz pre-amplifier (RSA5103B/RSA5106B only)
51	1 MHz to 315/26.5 GHz pre-amplifier (RSA5115B/RSA5126B only)
53	Memory 4 GB acquisition memory total

Table continued...

Option	Description
56	Removable hard disk drive (not compatible with Option 59)
59	Internal hard disk drive, incompatible with Option 56
65	Digital I and Q outputs
66	Zero-span output
6566	Digital I and Q outputs and Zero-span output
PFR	Precision frequency reference
PFR50	Precision frequency reference and internal preamp (RSA5103B/ RSA5126B only)
SSD	Includes additional solid-state drive for units equipped with Option 56. Minimum disk capacity is 480 GB. The Microsoft Windows 7 operating system and instrument software is preinstalled.

### Examples

\*OPT? might return 10,21,22, indicating that Option 10, 21, and 22 are currently installed in the analyzer.

## Output commands

### OUTPut:ETHernet:ADdResS

Sets or queries the IPv4 address that is used to connect with the client PC for 40 GbE streaming.

### Conditions

Only valid for RSA7100A instruments with option STREAMNL-SVPC installed.

### Group

Output commands

### Syntax

```
OUTPut:ETHernet:ADdResS "###.###.###.###"
```

```
OUTPut:ETHernet:ADdResS?
```

### Related Commands

### Arguments

"###.###.###.###" is a 12 digit IPv4 address.

### Returns

"###.###.###.###" is the 12 digit IPv4 address.

---

## Examples

OUTPUT:ETHERNET:ADDRESS 063.111.011.222 sets the IPv4 address to the given address.

## OUTPut:ETHernet:ENABLE

Sets or queries the status of 40 Gb Ethernet streaming.

### Conditions

Only valid for RSA7100A instruments with option STREAMNL-SVPC installed.

### Group

Output commands

### Syntax

```
OUTPut:ETHernet:ENABLE {1 | 0}
```

```
OUTPut:ETHernet:ENABLE?
```

### Related Commands

### Arguments

1 sets the device to streaming.

0 sets the device to not stream.

### Returns

1 indicates that the device is currently streaming.

0 indicates that the device is not streaming.

### Examples

OUTPUT:ETHERNET:ENABLE 1 sets the device to stream through 40 GbE.

## OUTPut:ETHernet:PORT

Sets or queries the port number that is used to connect with the client PC when using 40 GbE.

### Conditions

Only valid for RSA7100A instruments with option STREAMNL-SVPC installed.

### Group

Output commands

### Syntax

```
OUTPut:ETHernet:PORT <NR1>
```

OUTPut:ETHernet:PORT?

## Related Commands

### Arguments

<NR1> is an integer based value with a range between 1 and 65535.

### Returns

Integer based value with a range between 1 and 65535.

### Examples

OUTPUT:ETHERNET:PORT? might return 22, indicating that port 22 is being used to connect with the client PC.

## OUTPut:IF:{BANDwidth|BWIDth}

Sets or queries the IF output filter when OUTPut:IF[:STATe] is set to ON.

### Conditions

Measurement views: All (Option 05 only)

### Group

Output commands

### Syntax

```
OUTPut:IF:{BANDwidth|BWIDth} { FLATtop | GAUSSian }
```

```
OUTPut:IF:{BANDwidth|BWIDth}?
```

### Arguments

FLATtop selects the flattop filter.

GAUSSian selects the Gaussian filter.

### Examples

OUTPUT:IF:BANDWIDTH GAUSSIAN selects the Gaussian filter for the IF output.

## OUTPut:IF[:STATe]

Determines whether to turn on or off the 500 MHz IF Out on the rear panel.

### Conditions

Measurement views: All (Option 05 only)

### Group

Output commands

## Syntax

```
OUTPut:IF[:STATe] { OFF | ON | 0 | 1 }
```

```
OUTPut:IF[:STATe] ?
```

## Related Commands

```
OUTPut:IF:{BANDwidth|BWIDth}
```

## Arguments

OFF or 0 turns off IF Out.

ON or 1 turns on IF Out.

## Examples

```
OUTPUT:IF:STATE ON
```

turns on IF Out.

## OUTPut:IQ:SOURce

Sets or queries the IQ data transfer state via the LVDS card in the CTRL7100A with option 05.

## Conditions

Measurement views: All (Option 05 only for RSA7100A)

## Group

Output commands

## Syntax

```
OUTPut:IQ:SOURce { OFF | NORMal | CTP }
```

```
OUTPut:IQ:SOURce ?
```

## Arguments

OFF sets the IQ data transfer via LVDS to Off. This means that no streaming of data is occurring.

NORMAL sets the IQ data transfer via LVDS to On. This means that streaming of data, if available, is occurring.

CTP sets the IQ data transfer via LVDS to Test (Count Pattern). This means the system is validating that the LVDS system connections are correct by sending a known counting pattern from the CTRL7100A LVDS card to the receiving LVDS card. Check the counting pattern on the receiving end to verify valid data transfer.

## Examples

```
OUTPUT:IQ:SOURCE?
```

might return NORMAL, indicating that IQ data transfer via LVDS is set to On and occurring if data is available.

## OUTPut:IQ[:STATe]

Determines whether to enable or disable the digital IQ output data stream from the rear panel connectors.

## Conditions

Measurement views: All (Option 05 only)

## Group

Output commands

## Syntax

```
OUTPut:IQ[:STATe] { 0 | 1 | OFF | ON }
```

```
OUTPut:IQ[:STATe]?
```

## Arguments

OFF or 0 disables the digital IQ output.

ON or 1 enables the digital IQ output.

At \*RST, this value is set to OFF.

## Examples

OUTP:IQ:STAT ON enables the digital IQ output.

OUTP:IQ:STAT? may return OFF or 0, which means the digital IQ output is disabled.

## OUTPut:NOISe[:STATe]

Determines whether to turn on or off the +28 V DC Out on the rear panel.

## Conditions

Measurement views: All

## Group

Output commands

## Syntax

```
OUTPut:NOISe[:STATe] { OFF | ON | 0 | 1 }
```

```
OUTPut:NOISe[:STATe]?
```

## Arguments

OFF or 0 turns off +28 V DC Out.

ON or 1 turns on +28 V DC Out.

## Examples

OUTPUT:NOISE:STATE ON turns on +28 V DC Out.



## OUTPut:TGen:BRIDge:MODE

Sets or queries the tracking generator bridge mode.

### Conditions

Requires Option 04 in a RSA500A series or RSA600A series instrument.

### Group

Output commands

### Syntax

```
OUTPut:TGen:BRIDge:MODE { REFerence | CROSSover | REFlection| DISabled }
```

```
OUTPut:TGen:BRIDge:MODE
```

### Arguments

### Returns

### Examples

```
OUTPUT:TGEN:BRIDGE:MODE
```

## OUTPut:TGen:ENABLE

Sets or queries the power state of the tracking generator.

### Conditions

Requires Option 04 in a RSA500A series or RSA600A series instrument.

### Group

Output commands

### Syntax

```
OUTPut:TGen:ENABLE { OFF | ON | 0 | 1 }
```

```
OUTPut:TGen:ENABLE?
```

### Related Commands

### Arguments

### Returns

### Examples

```
OUTPUT:TGEN:ENABLE
```

## **OUTPut:TGen:FREQuency**

Sets or queries the tracking generator frequency in Hz. The spectrum analyzer center frequency will also be set to match (track) the tracking generator frequency.

### **Conditions**

Requires Option 04 in a RSA500A series or RSA600A series instrument.

### **Group**

Output commands

### **Syntax**

```
OUTPut:TGen:FREQuency <value>
```

```
OUTPut:TGen:FREQuency?
```

### **Arguments**

### **Returns**

### **Examples**

```
OUTPUT:TGEN:FREQUENCY
```

## **OUTPut:TGen:INSTalled? (Query Only)**

Queries for the presence of the tracking generator hardware.

### **Conditions**

Requires Option 04 in a RSA500A series or RSA600A series instrument.

### **Group**

Output commands

### **Syntax**

```
OUTPut:TGen:INSTalled?
```

### **Returns**

### **Examples**

```
OUTPUT:TGEN:INSTALLED?
```

## **OUTPut:TGen:LEVel**

Sets or queries the tracking generator output level in dBm.

## Conditions

Requires Option 04 in a RSA500A series or RSA600A series instrument.

## Group

Output commands

## Syntax

```
OUTPut:TGen:LEVel <value>
```

```
OUTPut:TGen:LEVel?
```

## Related Commands

## Arguments

## Returns

## Examples

```
OUTPUT:TGEN:LEVEL
```

## OUTPut:USERapi:ENABLE

Sets or queries the status for streaming API.

## Conditions

Only valid for RSA7100A instruments with option CUSTOM-APINL-SVPC installed.

## Group

Output commands

## Syntax

```
OUTPut:USERapi:ENABle {1 | 0}
```

```
OUTPut:USERapi:ENABle?
```

## Related Commands

## Arguments

1 sets the device to streaming.

0 sets the device to not stream.

## Returns

1 indicates that the device is currently streaming.

0 indicates that the device is not streaming.

## Examples

OUTPUT : USERAPI : ENABLE 0 sets the device to not stream through API.

## OUTPut:ZS:{BANDwidth|BWIDth}

Sets or queries the resolution bandwidth RBW filter for the Zero Span output on the rear panel connector.

### Conditions

Measurement views: DPX, Spectrum

### Group

Output commands

### Syntax

```
OUTPut:ZS:{BANDwidth|BWIDth} <value>
```

```
OUTPut:ZS:{BANDwidth|BWIDth}?
```

### Arguments

<value>::<NRf> specifies the resolution bandwidth.

## OUTPut:ZS:{BANDwidth|BWIDth}:AUTO

Determines whether to set the Zero Span resolution bandwidth manually or automatically.

### Conditions

Measurement views: DPX, Spectrum

### Group

Output commands

### Syntax

```
OUTPut:ZS:{BANDwidth|BWIDth}:AUTO { OFF | ON | 0 | 1 }
```

```
OUTPut:ZS:{BANDwidth|BWIDth}:AUTO?
```

### Arguments

OFF or 0 manually sets the Zero Span output.

ON or 1 automatically sets the Zero Span output.

## Examples

OUTPUT : ZS : BANDWIDTH : AUTO ON specifies that the Zero Span output will be set automatically.

## OUTPut:ZS:{BANDwidth|BWIDth}:STATe

Sets or queries the state of the resolution bandwidth filter for the output connector on the rear panel.

### Conditions

Measurement views: DPX, Spectrum

### Group

Output commands

### Syntax

```
OUTPut:ZS:{BANDwidth|BWIDth}:STATe { Auto | Manual | None }
```

```
OUTPut:ZS:{BANDwidth|BWIDth}:STATe?
```

### Arguments

Auto sets the resolution bandwidth filter to automatic mode.

Manual sets the resolution bandwidth filter to manual mode.

None bypasses the resolution bandwidth filter.

## OUTPut:ZS:DELay? (Query Only)

Queries the delay from the input connector to the Zero Span output. The delay includes the propagation time through the entire system including filters and digital processing.

### Conditions

Measurement views: DPX, Spectrum

### Group

Output commands

### Syntax

```
OUTPut:ZS:DELay?
```

### Arguments

None

### Returns

<NRf> The actual delay of the signal through the system.

## OUTPut:ZS:FILTer:STATe

Sets or queries the signal path of the filter to the Zero Span output.

## Conditions

Measurement views: DPX, Spectrum

## Group

Output commands

## Syntax

```
OUTPut:ZS:FILTer:STATe { AUTO | HIGH | LOW }
```

```
OUTPut:ZS:FILTer:STATe?
```

## Arguments

AUTO The filter path is calculated internally.

HIGH Uses the high-band 10 MHz filter.

LOW Uses the low-band 100 kHz filter.

## OUTPut:ZS[:STATe]

Determines whether to enable or disable the Zero Span output from the rear panel connector.

## Conditions

Measurement views: DPX, Spectrum

## Group

Output commands

## Syntax

```
OUTPut:ZS[:STATe] { OFF | ON | 0 | 1 }
```

```
OUTPut:ZS[:STATe]?
```

## Arguments

OFF or 0 disables the Zero Span output

ON or 1 enables the Zero Span output

## Examples

OUTPUT:ZS:STATE ON enables on the Zero Span output.

## OUTPut:ZS:VOLTage

Sets or queries the voltage level at the Zero Span output on the rear panel connector.

## Conditions

Measurement views: DPX, Spectrum

## Group

Output commands

## Syntax

```
OUTPut:ZS:VOLTage <value>
```

```
OUTPut:ZS:VOLTage?
```

## Arguments

<value>: : <NRf> voltage level at the Zero Span output connector.

## OUTPut:ZS:VOLTage:AUTO

Sets or queries or queries how the output voltage is referenced.

## Conditions

Measurement views: DPX, Spectrum

## Group

Output commands

## Syntax

```
OUTPut:ZS:VOLTage:AUTO { OFF | ON | 0 | 1 }
```

```
OUTPut:ZS:VOLTage:AUTO?
```

## Arguments

ON or 1 sets the output voltage referenced to 1.0 V.

OFF or 0 allows users to scale down the output voltage from 0.5 V to 1.0 V.

## Read commands

### READ:ACPower? (Query Only)

Returns the Channel power and ACPR measurement results for all available channels.

## Conditions

Measurement views: Channel power and ACPR

## Group

Read commands

## Syntax

```
READ:ACPower?
```

## Arguments

None

## Returns

<chan\_power>, <acpr\_lower(1)>, <acpr\_upper(1)>,  
<acpr\_lower(2)>, <acpr\_upper(2)>, ...  
<acpr\_lower(n)>, <acpr\_upper(n)>

Where

<chan\_power> is the average power of the main channel as the power reference in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

<acpr\_lower(n)> is the ACPR for the lower channel #n in dB.

<acpr\_upper(n)> is the ACPR for the upper channel #n in dB.

The number of n depends on the setting of the [SENSe]:ACPower:CHANnel:PAIRs command.

## Examples

READ:ACPOWER? might return 4.227,-28.420,-23.847,-22.316,-29.225, indicating

(average power of the main channel) = 4.227 dBm,

(ACPR for the lower channel 1) = -28.420 dB,

(ACPR for the upper channel 1) = -23.847 dB,

(ACPR for the lower channel 2) = -22.316 dB, and

(ACPR for the upper channel 2) = -29.225 dB.

## READ:ACPower:CHANnel:POWer? (Query Only)

Returns the average power of the main channel (power reference) in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Read commands

## Syntax

READ:ACPower:CHANnel:POWer?

## Arguments

None

## Returns

<chan\_power>::<NRf> is the average power of the main channel in dBm.



The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:ACPOWER:CHANNEL:POWER?` might return 4.227, indicating that the average power of the main channel is 4.227 dBm.

## READ:ACPower:SPECTrum? (Query Only)

Returns spectrum trace data of the Channel power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Read commands

### Syntax

`READ:ACPower:SPECTrum?`

### Arguments

None

### Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the spectrum trace data in dBm for the point `n`,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:ACPOWER:SPECTRUM?` might return `#43204xxxx...` (3204-byte data) for the spectrum trace data of the Channel power and ACPR measurement.

## READ:{AM|FM|PM}? (Query Only)

Returns the trace data in the AM/FM/PM measurement.

### Conditions

Measurement views: AM, FM, PM

## Group

Read commands

## Syntax

READ: {AM|FM|PM}?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the modulation factor in percent (AM), frequency deviation in Hz (FM), or phase deviation in degrees (PM) at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:AM might return #3156xxx... (156-byte data) for the AM measurement trace.

## READ:AM:AMIndex? (Query Only)

Returns the modulation index which is (positive peak modulation factor - negative peak modulation factor)/2, returned as a percentage (%).

## Conditions

Measurement views: AM

## Group

Read commands

## Syntax

READ:AM:AMIndex?

## Arguments

None

## Returns

<value>::<NRf> the modulation index.

---

## Examples

`READ:AM:AMINDEX?` might return 36.48, indicating the modulation index is 36.48%.

## READ:AM:AMNegative? (Query Only)

Returns the negative peak modulation factor (–AM) in the AM measurement.

### Conditions

Measurement views: AM

### Group

Read commands

### Syntax

`READ:AM:AMNegative?`

### Arguments

None

### Returns

`<–AM> : : <NRf>` is the negative peak modulation factor in percent (%).

## Examples

`READ:AM:AMNEGATIVE?` might return –23.4, indicating the negative peak modulation factor is –23.4%.

## READ:AM:AMPositive? (Query Only)

Returns the positive peak modulation factor (+AM) in the AM measurement.

### Conditions

Measurement views: AM

### Group

Read commands

### Syntax

`READ:AM:AMPositive?`

### Arguments

None

### Returns

`<+AM> : : <NRf>` is the positive peak modulation factor in percent (%).

## Examples

READ:AM:AMPOSITIVE? might return 43.8, indicating the positive peak modulation factor is 43.8%.

## READ:AM:RESult? (Query Only)

Returns the AM measurement results.

### Conditions

Measurement views: AM

### Group

Read commands

### Syntax

READ:AM:RESult?

### Arguments

None

### Returns

<+AM>, <-AM>, <Total AM>

Where

<+AM>: : <NRf> is the positive peak modulation factor in percent (%).

<-AM>: : <NRf> is the negative peak modulation factor in percent (%).

<Total AM>: : <NRf> is the (peak to peak modulation factor)/2 in percent (%).

## Examples

READ:AM:RESULT? might return 62.63,—50.89,56.76.

## READ:AUDio:FERRor? (Query Only)

Returns the carrier frequency error in the audio measurement. This command is only available for FM and PM measurements.

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

READ:AUDio:FERRor?

## Arguments

None

## Returns

`<value> ::= <NRf>` the carrier frequency error.

## Examples

`READ:AUDIO:FERROR?` might return 419.9529809622 indicating that the audio frequency error is approximately 420 Hz.

## READ:AUDio:FREQuency? (Query Only)

Returns the audio frequency.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

`READ:AUDio:FREQuency?`

## Arguments

None

## Returns

`<value> ::= <NRf>` the audio frequency.

## Examples

`READ:AUDIO:FREQUENCY?` might return 5.8239462705E+3 indicating that the audio frequency is 5.824 kHz.

## READ:AUDio:HARMonic:COUNT? (Query Only)

Returns the number of harmonics in the audio spectrum measurement. The number of harmonics appear in the results table in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

READ:AUDio:HARMonic:COUNT?

## Related Commands

READ:AUDio:NHARmonic:COUNT?

## Arguments

None

## Returns

<count> ::= <NR1> the number of harmonics in the measurement view.

## Examples

READ:AUDIO:HARMONIC:COUNT? might return 3 indicating that there are three harmonics in the measurement.

## READ:AUDio:HARMonic<x>:AMPLitude? (Query Only)

Returns the amplitude of the specified harmonic in the audio spectrum measurement.

The value <x> represents the specific harmonic, from 1 to the value specified by the CALCulate:AUDio:HARMonic:HNUMber command.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

READ:AUDio:HARMonic<x>:AMPLitude?

## Related Commands

CALCulate:AUDio:HARMonic:HNUMber

## Arguments

None

## Returns

<value> ::= <NRf> the amplitude of the specified harmonic in dBc.

## Examples

READ:AUDio:HARMonic3:AMPLitude? might return -2.861 indicating that amplitude of the third harmonic is -2.86 dBc.

## READ:AUDio:HARMonic<x>:FREQuency? (Query Only)

Returns the frequency of the specified harmonic in the audio spectrum measurement.

The value <x> represents the specific harmonic, from 1 to the value specified by the `CALCulate:AUDio:HARMonic:HNUMber` command.

### Conditions

Measurement views: Audio Spectrum

### Group

Read commands

### Syntax

```
READ:AUDio:HARMonic<x>:FREQuency?
```

### Related Commands

```
CALCulate:AUDio:HARMonic:HNUMber
```

### Arguments

None

### Returns

<value> ::= <NRF> the frequency of the specified harmonic.

### Examples

`READ:AUDio:HARMonic3:FREQuency?` might return `17.4718007813E+3` indicating that the frequency of the third harmonic is 17.472 kHz.

## READ:AUDio:HNOise? (Query Only)

Returns the difference between the current RMS modulation value and the reference value.

This information is only available when the Hum & Noise is measurement is enabled; see the `[SENSe]:AUDio:HNOise:ENABle` command. The reference value is stored after capturing the reference by pressing the Capture Reference button or by enabling the `[SENSe]:AUDio:HNOise:REFerence` command.

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

```
READ:AUDio:HNOise?
```

## Related Commands

[SENSe]:AUDio:HNOise:ENABle [SENSe]:AUDio:HNOise:REFerence

## Arguments

None

## Returns

<value> ::= <NRF> specifies the difference between the Hum & Noise reference and the actual signal in dB.

## Examples

READ:AUDIO:HNOISE? might return 0.00 dB indicating that there is no difference between the Hum & Noise reference and the actual signal.

## READ:AUDIO:HNREFERENCE? (Query Only)

Returns the Hum and Noise RMS modulation reference value.

The value is stored after capturing the reference by pressing the Capture Reference button or by enabling the [SENSe]:AUDio:HNOise:REFerence command. This information is only available when the Hum & Noise measurement is enabled; see the [SENSe]:AUDio:HNOise:ENABle command.

For AM signal types, the reference represents modulation depth. For FM signal types, the reference represents frequency deviation. For PM signal types the reference represents phase. For Direct signal types the reference represents power.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

READ:AUDIO:HNREFERENCE?

## Related Commands

[SENSe]:AUDio:HNOise:ENABle [SENSe]:AUDio:HNOise:REFerence

## Arguments

None

## Returns

<value> ::= <NRF> the Hum & Noise reference frequency.

## Examples

READ:AUDIO:HNREFERENCE? might return 20.581121E+3 indicating that the Hum & Noise reference frequency is 20.58 kHz.



## READ:AUDio:HPTPeak? (Query Only)

Returns the half peak-to-peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

```
READ:AUDio:HPTPeak?
```

### Arguments

None

### Returns

<value> ::= <NRF> specifies the half peak modulation excursion.

### Examples

READ:AUDIO:HPTPEAK? might return 125.28231E+3 indicating that the half peak modulation excursion is 125.28 kHz.

## READ:AUDio:MODDist? (Query Only)

Returns the modulation distortion for the audio measurement.

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

```
READ:AUDio:MODDist?
```

### Arguments

None

### Returns

<value> ::= <NRf> specifies the modulation distortion.

## Examples

`READ:AUDIO:MODDIST?` might return 98.6282113 indicating that the modulation distortion is 98.628%.

## **READ:AUDio:NHARmonic:COUNT? (Query Only)**

Returns the number of non-harmonics in the audio spectrum measurement.

The number of non-harmonics appear in the results table in the audio spectrum view.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

`READ:AUDio:NHARmonic:COUNT?`

## Related Commands

`READ:AUDio:HARMonic:COUNT?`

## Arguments

None

## Returns

`<count>` ::= `<NR1>` the number of non-harmonics in the measurement view.

## Examples

`READ:AUDIO:NHARMONIC:COUNT?` might return 7 indicating that there are seven harmonics in the measurement.

## **READ:AUDio:NHARmonic<x>:AMPLitude? (Query Only)**

Returns the amplitude of the specified non-harmonic in the audio spectrum measurement.

The value `<x>` represents the specific non-harmonic, from 1 to the value specified by the `CALCulate:AUDio:HARMonic:NHNumber` command.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

`READ:AUDio:NHARmonic<x>:AMPLitude?`

## Related Commands

`CALCulate:AUDio:HARMonic:NHNumber`

## Arguments

None

## Returns

`<value> ::= <NRF>` the amplitude of the specified non-harmonic in dBc.

## Examples

`READ:AUDio:NHARMonic3:AMPLitude?` might return -2.861 indicating that amplitude of the third non-harmonic is -2.86 dBc.

## READ:AUDio:NHARmonic<x>:FREQuency? (Query Only)

Returns the frequency of the specified non-harmonic in the audio spectrum measurement.

The value `<x>` represents the specific non-harmonic, from 1 to the value specified by the `CALCulate:AUDio:HARMonic:NHNumber` command.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

`READ:AUDio:NHARmonic<x>:FREQuency?`

## Related Commands

`CALCulate:AUDio:HARMonic:NHNumber`

## Arguments

None

## Returns

`<value> ::= <NRF>` the frequency of the specified non-harmonic.

## Examples

`READ:AUDio:NHARMonic3:FREQuency?` might return 17.4718007813E+3 indicating that the frequency of the third non-harmonic is 17.472 kHz.

## READ:AUDio:PNEGative? (Query Only)

Returns the minus peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

```
READ:AUDio:PNEGative?
```

## Arguments

None

## Returns

## Examples

`READ:AUDIO:PNEGATIVE?` might return `-196.04321E+3` indicating that the minus peak modulation excursion is `-196.04` kHz.

## READ:AUDio:POWer? (Query Only)

Returns the carrier power or signal power depending the signal type.

If the signal type is Direct, the returned value is the average power of the input signal. If the signal type is AM, FM, or PM, the returned value is the carrier power; the average power of the carrier signal with the modulation removed.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

```
READ:AUDio:POWer?
```

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the carrier or signal power in dBm.

## Examples

`READ:AUDIO:POWER?` might return `-22.231123` indicating that the carrier power is `-22.23` dBm.

## READ:AUDio:PPOSitive? (Query Only)

Returns the positive peak modulation excursion (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

`READ:AUDio:PPOSitive?`

## Arguments

None

## Returns

`<value> ::= <Nrf>` specifies the positive peak modulation excursion.

## Examples

`READ:AUDIO:PPOSITIVE?` might return `215.04321E+3` indicating that the positive peak modulation excursion is `-215.04` kHz.

## READ:AUDio:RMS? (Query Only)

Returns the RMS modulation excursion for the audio measurement (where the modulation excursion depends on the signal type).

For AM signal types, the modulation excursion is "% Modulation Depth." For FM signal types, the modulation excursion is "Frequency Deviation." For PM signal types, the modulation excursion is "Phase Deviation." For Direct signal types, there is no modulation excursion; it is actually "signal excursion."

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

`READ:AUDio:RMS?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the RMS modulation excursion.

## Examples

`READ:AUDIO:RMS?` might return `20.575039E+3` indicating that RMS modulation excursion is 20.575 kHz.

## READ:AUDio:SINad? (Query Only)

Returns the signal-to-noise and distortion for the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

`READ:AUDio:SINad?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the RMS modulation excursion.

## Examples

`READ:AUDIO:SINAD?` might return `176.229024E-3` indicating that the signal-to-noise and distortion for the audio measurement is 0.18 dB.

## READ:AUDio:SNOise? (Query Only)

Returns the signal level to noise level with the harmonic distortion and non-harmonic distortion components removed.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

READ:AUDio:SNOise?

## Arguments

None

## Returns

<value> ::= <Nrf> specifies the signal level to noise level.

## Examples

READ:AUDIO:SNOISE? might return -12.8156364 indicating the signal level to noise level is -12.82 dB.

## READ:AUDio:SPECTrum:TRACe<x>? (Query Only)

Returns the audio spectrum trace data for the audio measurement.

The <x> represents trace 1 for this measurement. No other traces are supported.

## Conditions

Measurement views: Audio Spectrum

## Group

Read commands

## Syntax

READ:AUDio:SPECTrum:TRACe<x>?

## Arguments

None

## Returns

#<num\_digit><num\_byte><date(1)><data(2)> . . . <data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating point format specified in IEEE.488.2.

## Examples

READ:AUDio:SPECTrum:TRACe1? might return #516004xxxx ... (16004 byte data) for the audio measurement.

## READ:AUDio:THDist? (Query Only)

Returns the percentage of the total harmonic distortion in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

```
READ:AUDio:THDist?
```

## Related Commands

```
READ:AUDio:THDist:DB?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the percentage of the total harmonic distortion.

## Examples

READ:AUDIO:THDIST? might return 53.332921 indicating the percentage of total harmonic distortion is 53.333%.

## READ:AUDio:THDist:DB? (Query Only)

Returns the total harmonic distortion in dB in the audio measurement.

## Conditions

Measurement views: Audio Summary

## Group

Read commands

## Syntax

```
READ:AUDio:THDist:DB?
```

## Related Commands

```
READ:AUDio:THDist?
```

## Arguments

None

## Returns

<value> ::= <NRf> specifies the total harmonic distortion in dB.



## Examples

`READ:AUDIO:THDIST:DB?` might return `-5.46009` indicating that the total harmonic distortion is `-5.46` dB.

## READ:AUDio:TNHDist? (Query Only)

Returns the percentage of the total non-harmonic distortion in the audio measurement.

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

`READ:AUDio:TNHDist?`

### Related Commands

`READ:AUDio:TNHDist:DB?`

### Arguments

None

### Returns

`<value>` ::= `<NRf>` specifies the percentage of the total non-harmonic distortion.

## Examples

`READ:AUDIO:TNHDIST?` might return `297.332921` indicating the percentage of total non-harmonic distortion is `297.333%`.

## READ:AUDio:TNHDist:DB? (Query Only)

Returns the total non-harmonic distortion in dB in the audio measurement.

### Conditions

Measurement views: Audio Summary

### Group

Read commands

### Syntax

`READ:AUDio:TNHDist:DB?`

### Related Commands

`READ:AUDio:TNHDist?`

## Arguments

None

## Returns

`<value> ::= <NRf>` specifies the total non-harmonic distortion in dB.

## Examples

`READ:AUDIO:TNHDIST:DB?` might return 9.46009 indicating that the total harmonic distortion is 9.46 dB.

## **READ:AVTime[:MEASview<y>]:AVERage? (Query Only)**

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SELEct` command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

`READ:AVTime[:MEASview<y>]:AVERage?`

## Arguments

The parameter `<y>=1` to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Returns

`<avg> ::= <NRf>` is the RMS amplitude in dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:AVTIME:MEASVIEW1:AVERAGE?` might return -2.53, indicating the RMS amplitude is -2.53 dBm on Amplitude versus Time Measurement View 1.

## **READ:AVTime[:MEASview<y>]:{FIRSt|SECond|THIRd|FOURth}? (Query Only)**

Returns the trace data in the Amplitude versus Time measurement.

The mnemonics `FIRSt`, `SECond`, `THIRd`, and `FOURth` represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the `TRACe<x>:AVTime[:MEASview<y>]` command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

READ:AVTIME:MEASVIEW1:FIRST? might return #3156xxx... (156-byte data) for Trace 1 on Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MAXimum?
```

## Related Commands

```
READ:AVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max>::<Nrf> is the maximum Amplitude in dBm.

The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

READ:AVTIME:MEASVIEW1:MAXIMUM? might return -2.84, indicating the maximum amplitude is -2.84 dBm for Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MAXLocation?
```

## Related Commands

```
READ:AVTime[:MEASview<y>]:MAXimum?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max\_time>::<Nrf> is the time at the maximum in seconds.

## Examples

READ:AVTIME:MEASVIEW1:MAXLOCATION? might return 25.03E-9, indicating the amplitude is maximum at 25.03 ns on Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MINimum? (Query Only)

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MINimum?
```

## Related Commands

```
READ:AVTime[:MEASview<y>]:MINLocation?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<min>::<Nrf> is the minimum amplitude in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:AVTIME:MEASVIEW1:MINIMUM? might return -57.64, indicating the minimum amplitude is -57.64 dBm on Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MINLocation? (Query Only)

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
READ:AVTime[:MEASview<y>]:MINimum?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

`<min_time>::<NRf>` is the time at the minimum in seconds.

## Examples

`READ:AVTIME:MEASVIEW1:MINLOCATION?` might return `450.7E-9`, indicating the amplitude is minimum at 450.7 ns on Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:RESult? (Query Only)

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

`READ:AVTime[:MEASview<y>]:RESult?`

## Arguments

The parameter `<y>=1` to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

## Returns

`<max>,<max_time>,<min>,<min_time>,<rms>`

Where

`<max>::<NRf>` is the maximum amplitude in dBm.

`<max_time>::<NRf>` is the time at the maximum in seconds.

`<min>::<NRf>` is the minimum amplitude in dBm.

`<min_time>::<NRf>` is the time at the minimum in seconds.

`<rms>::<NRf>` is the RMS amplitude in dBm.

The unit of amplitude can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:AVTIME:MEASVIEW1:RESULT?` might return

`-2.68,48.62E-6,-82.47,22.11E-6,-8.24`, indicating that

the maximum amplitude is -2.68 dBm at 48.62  $\mu$ s,

the minimum amplitude is -82.47 dBm at 22.11  $\mu$ s, and

the RMS amplitude is -8.24 dBm on Amplitude versus Time Measurement View 1.

## **READ:AVTime[:MEASview<y>]:AVERage:{FIRSt|SECond|THIRd|FOURth}?**

Returns the RMS (root-mean-square) value for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

The mnemonics `FIRSt`, `SECond`, `THIRd`, and `FOURth` represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the `TRACe<x>:AVTime[:MEASview<y>]` command subgroup.

### **Conditions**

Measurement views: Amplitude versus Time

### **Group**

Read commands

### **Syntax**

```
READ:AVTime[:MEASview<y>]:AVERage:{FIRSt|SECond|THIRd|FOURth}?
```

### **Related commands**

```
READ:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?
```

### **Arguments**

The parameter `<y>`=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

### **Returns**

`<avg>: : <Nrf>` is the RMS amplitude in dBm.

The unit can be changed by the `[SENSE]:POWER:UNITs` command.

### **Examples**

`READ:AVTIME:MEASVIEW1:AVERAGE:FIRST?` might return -2.53, indicating the RMS amplitude is -2.53 dBm on Trace1 for Amplitude versus Time Measurement View 1.

## **READ:AVTime[:MEASview<y>]:MAXimum:{FIRSt|SECond|THIRd|FOURth}?**

Returns the maximum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the `TRACe<x>:AVTime[:MEASview<y>]:SElect` command.

The mnemonics `FIRSt`, `SECond`, `THIRd`, and `FOURth` represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the `TRACe<x>:AVTime[:MEASview<y>]` command subgroup.

### **Conditions**

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MAXimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Related commands

```
READ:AVTime[:MEASview<y>]:MAXLocation:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<max>: : <Nrf> is the maximum Amplitude in dBm.

The unit can be changed by the [:SENSe]:POWer:UNITs command.

## Examples

READ:AVTIME:MEASVIEW1:MAXIMUM:FIRST? might return -2.84, indicating the maximum amplitude is -2.84 dBm on Trace1 for Amplitude versus Time Measurement View 1.

## **READ:AVTime[:MEASview<y>]:MAXLocation:{FIRSt|SECond|THIRd|FOURth}?**

Returns the time at which the amplitude is maximum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SELEct command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime[:MEASview<y>] command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MAXLocation:{FIRSt|SECond|THIRd|FOURth}?
```

## Related commands

```
READ:AVTime[:MEASview<y>]:MAXimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1



## Returns

<max>: : <NRf> is the maximum Amplitude in dBm.

## Examples

READ:AVTIME:MEASVIEW1:MAXLOCATION:FIRST? might return 25.03E-9, indicating the amplitude is maximum at 25.03 ns on Trace1 for Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?

Returns the minimum value for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SELEct command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime[:MEASview<y>] command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

READ:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?

## Related commands

READ:AVTime[:MEASview<y>]:MINLocation:{FIRSt|SECond|THIRd|FOURth}?

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<min>: : <NRf> is the minimum amplitude in dBm. The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

READ:AVTIME:MEASVIEW1:MINIMUM:FIRST? might return -57.64, indicating the minimum amplitude is -57.64 dBm on Trace1 for Amplitude versus Time Measurement View 1.

## READ:AVTime[:MEASview<y>]:MINLocation:{FIRSt|SECond|THIRd|FOURth}?

Returns the time at which the amplitude is minimum for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SELEct command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime[:MEASview<y>] command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:MINLocation:{FIRSt|SECond|THIRd|FOURth}?
```

## Related commands

```
READ:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<min>::<Nrf> is the minimum amplitude in dBm. The unit can be changed by the [:SENSe]:POWer:UNITs command.

## Examples

```
READ:AVTIME:MEASVIEW1:MINLOCATION:FIRST? might return 450.7E-9, indicating the amplitude is minimum at 450.7 ns on Trace1 for Amplitude versus Time Measurement View 1.
```

## **READ:AVTime[:MEASview<y>]:RESult:{FIRSt|SECond|THIRd|FOURth}?**

Returns the measurement results for the selected trace in the Amplitude versus Time measurement. Select the trace using the TRACe<x>:AVTime[:MEASview<y>]:SElect command.

The mnemonics FIRSt, SECond, THIRd, and FOURth represent Trace 1, Trace 2, Trace 3, and Math trace, respectively. The traces can be specified by the TRACe<x>:AVTime[:MEASview<y>] command subgroup.

## Conditions

Measurement views: Amplitude versus Time

## Group

Read commands

## Syntax

```
READ:AVTime[:MEASview<y>]:RESult:{FIRSt|SECond|THIRd|FOURth}?
```

## Related commands

```
READ:AVTime[:MEASview<y>]:MINimum:{FIRSt|SECond|THIRd|FOURth}?
```

## Arguments

The parameter  $\langle y \rangle = 1$  to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default  $y$  is 1

## Returns

$\langle \text{max} \rangle, \langle \text{max\_time} \rangle, \langle \text{min} \rangle, \langle \text{min\_time} \rangle, \langle \text{rms} \rangle$

Where

$\langle \text{max} \rangle :: \langle \text{NRf} \rangle$  is the maximum amplitude in dBm.

$\langle \text{max\_time} \rangle :: \langle \text{NRf} \rangle$  is the time at the maximum in seconds.

$\langle \text{min} \rangle :: \langle \text{NRf} \rangle$  is the minimum amplitude in dBm.

$\langle \text{min\_time} \rangle :: \langle \text{NRf} \rangle$  is the time at the minimum in seconds.

$\langle \text{rms} \rangle :: \langle \text{NRf} \rangle$  is the RMS amplitude in dBm.

The unit of amplitude can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:AVTIME:MEASVIEW1:RESULT:FIRST?` might return

`-2.68,48.62E-6,-82.47,22.11E-6,-8.24`, indicating that

the maximum amplitude is -2.68 dBm at 48.62  $\mu\text{s}$ ,

the minimum amplitude is -82.47 dBm at 22.11  $\mu\text{s}$ , and

the RMS amplitude is -8.24 dBm on Trace1 for Amplitude versus Time Measurement View 1.

## READ:BIBEmissions:FTX? (Query Only)

Returns the band number corresponding to the transmitted FTX detected in the Bluetooth InBand Emission display.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Read commands

## Syntax

`READ:BIBEmissions:FTX?`

## Arguments

None

## Returns

$\langle \text{value} \rangle ::= \langle \text{NR1} \rangle$  the band number.

## Examples

`READ:BIBEMISSIONS:FTX?` might return 39 indicating the band number corresponding to transmitted FTX.

## READ:BIBEmissions:POWer? (Query Only)

Returns all channel power results from the Bluetooth InBand Emission display.

### Conditions

Measurement views: Bluetooth InBand Emission

### Group

Read commands

### Syntax

`READ:BIBEmissions:POWer?`

### Arguments

None

### Returns

`<Band0_power>, <Band1_power>, <Band2_power>, . . . <Band78_power>`

Where:

`Bandx_power` is the integrated power in that band. All power values are in dBm

### Examples

`READ:BIBEMISSIONS:POWER?` might return 4.227,-28.420,-23.847, . . . ,-29.225, indicating: (integrated power for Band 0) = 4.227 dBm,

(Integrated power for Band 1) = -28.420 dBm,

(Integrated power for Band 2) = -23.847 dBm, and

(Integrated power for Band 78) = -29.225 dBm.

## READ:BIBEmissions:RESUlts:STATUs? (Query Only)

Returns the Pass/Fail result status of the Bluetooth InBand Emission display.

### Conditions

Measurement views: Bluetooth InBand Emission

### Group

Read commands

## Syntax

```
READ:BIBEmISSIONS:RESUlts:STATUs?
```

## Arguments

None

## Examples

READ:BIBEMISSIONS:RESULTS:STATUS? returns the Pass/Fail status result on the display.

## READ:BLUETOOTH:CONStE:FERRor? (Query Only)

Queries the frequency error in Hz in the Bluetooth Constellation. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:CONStE:FERRor?
```

## Arguments

None

## Returns

<freq\_error> ::= <Nrf> the frequency error.

## Examples

READ:BLUETOOTH:CONStE:FERRor? Might return -10.7E+3, which is a frequency error of -10.7 kHz.

## READ:BLUETOOTH:CONStE:TRACe? (Query Only)

Queries the Bluetooth Constellation trace data.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Read commands

## Syntax

READ:BLUEtooth:CONStE:TRACe?

## Arguments

None

## Returns

#<num\_digit><num\_byte><I(1)><Q1><I(2)><Q2> . . . <I(n)><Qn>

Where:

<num\_digit> is the number of bytes in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<I(n)> and <Qn> are the normalized I- and Q-coordinate values at the nth data point. 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:BLUEtooth:CONStE:TRACe? might return #43848xxxx . . . (3848-byte data) for Bluetooth Constellation trace.

## READ:BLUEtooth:EDIagram:FDEVIation? (Query Only)

Returns the frequency deviation vs. time trace data with the X values in the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth eye diagram.

## Group

Read commands

## Syntax

READ:BLUEtooth:EDIagram:FDEVIation?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<Y(1)> and <X(1)> are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

`READ:BLUETOOTH:EDIagram:FDEVIiation?` might return `#3160xxx...` (160-byte data) for the frequency deviation versus time trace.

## READ:BLUETOOTH:EDIagram:FERRor? (Query Only)

Returns the frequency error in the Bluetooth eye diagram measurement.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Read commands

### Syntax

`READ:BLUETOOTH:EDIagram:FERRor?`

### Arguments

None

### Returns

`<freq_error>` ::= `<NRf>` the frequency error in Hz.

## Examples

`READ:BLUETOOTH:EDIagram:FERRor?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

## READ:BLUETOOTH:EDIagram:HORiz:SCALE? (Query Only)

Returns the value of the horizontal scale in the Bluetooth eye diagram.

### Conditions

Measurement views: Bluetooth eye diagram.

### Group

Read commands

### Syntax

`READ:BLUETOOTH:EDIagram:HORiz:SCALE?`

### Arguments

None

## Returns

<value> ::= <NR3> the horizontal scale value.

## Examples

READ:BLUETOOTH:EDIAGRAM:HORIZ:SCALE? might return 2 indicating the horizontal scale value is 2 symbols.

## READ:BLUETOOTH:EDIAGRAM:I? (Query Only)

Returns the I vs. time trace in the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth eye diagram.

## Group

Read commands

## Syntax

READ:BLUETOOTH:EDIAGRAM:I?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<Y(1)> and <X(1)> are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

READ:BLUETOOTH:EDIAGRAM:I? might return #3160xxx... (160-byte data) for the I vs. time trace.

## READ:BLUETOOTH:EDIAGRAM:Q? (Query Only)

Returns the Q vs. time trace in the Bluetooth eye diagram.

## Conditions

Measurement views: Bluetooth eye diagram.

## Group

Read commands



## Syntax

READ:BLUEtooth:EDIagram:Q?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><Y(1)><X(1)><Y(2)><X(2)> . . . <Y(n)><X(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<Y(1)> and <X(1)> are the normalized I- and Q-coordinate values at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

READ:BLUEtooth:EDIagram:Q? might return #3160xxxx... (160-byte data) for the Q vs. time trace.

## READ:BLUEtooth:FDVTime:TRACe? (Query Only)

Returns the Bluetooth Frequency Deviation vs. Time trace data.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

READ:BLUEtooth:FDVTime:TRACe?

## Arguments

None

## Returns

#<num\_digits><num\_bytes><data(1)><data(2)> . . . <data(n)>

Where

<num\_digits> is the number of bytes in <num\_bytes>.

<num\_bytes> is the number of bytes of data that follow.

<data(n)> is the frequency deviation in Hz at the nth datapoint. The 4-byte little endian floating point format is specified in IEEE 488.2.

## Examples

`READ:BLUETOOTH:FDVTime:TRACe?` might return #3160xxx... (160-byte data) for the trace.

## **READ:BLUETOOTH:FREQuency:ERROr? (Query Only)**

Returns the frequency error from the Bluetooth Constellation diagram.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Read commands

### Syntax

`READ:BLUETOOTH:FREQuency:ERROr?`

### Arguments

None

### Returns

<NR3>, the frequency error value in Hz.

## Examples

`READ:BLUETOOTH:FREQUENCY:ERROR?` might return 0.0000 indicating the frequency error is 0.0000 Hz.

## **READ:BLUETOOTH:FREQuency:ERROr:TYPE**

Returns the frequency error type in the Bluetooth Constellation display.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Read commands

### Syntax

`READ:BLUETOOTH:FREQuency:ERROr:TYPE`

### Arguments

None

### Returns

1 indicates the error is automatically detected.

0 indicates the error is not automatically detected.

## Examples

`READ:BLUETOOTH:FREQUENCY:ERROR:TYPE` might return 1 indicating the frequency error is automatically detected.

## **READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:F1FZERO? (Query Only)**

Returns the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload— the frequency offset calculated in the preamble) in the Bluetooth Frequency Deviation vs. Time display.

### Conditions

Measurement views: Frequency Deviation vs. Time

### Group

Read commands

### Syntax

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:F1FZERO?`

### Arguments

None

### Returns

`<NR3>` the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble)

## Examples

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:F1FZERO?` might return -207.6465301514 indicating the drift in frequency offset ( $f_1 - f_0$  — the frequency offset calculated in the first interval in the payload — the frequency offset calculated in the preamble).

## **READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFN5? (Query Only)**

Returns the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away in the Bluetooth Frequency Deviation vs. Time display.

### Conditions

Measurement views: Frequency Deviation vs. Time

### Group

Read commands

### Syntax

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFN5?`

## Arguments

None

## Returns

<NR3> the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away.

## Examples

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNF5?` might return -207.6465301514 indicating the maximum drift of the frequency offset in payload intervals spaced 50  $\mu$ s away.

## **READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNF5:INTERVAL? (Query Only)**

Returns the interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNF5:INTERVAL?`

## Arguments

None

## Returns

<NR1> interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred.

## Examples

`READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNF5:INTERVAL?` might return 16, indicating the interval at which the maximum drift of frequency offset in a duration of 50  $\mu$ s occurred. Here 16 refers to the 16th interval and that the drift of frequency offset calculated in the 16th interval from that calculated in the 11th interval is the maximum among those calculated 50  $\mu$ s away.

## **READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO? (Query Only)**

Returns the maximum drift of the frequency offset of the intervals in the payload from the preamble packet in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO?
```

## Arguments

None

## Returns

<NR3> the maximum drift of frequency offset calculated in payload intervals from the offset calculated in the preamble.

## Examples

READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO? might return -207.6465301514, indicating that the maximum drift of frequency offset calculated in payload intervals from the offset calculated in the preamble is -207.6465301515 Hz.

## READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO:INTERVAL? (Query Only)

Returns the interval (n) at which the maximum drift  $f_n - f_0$  occurred in the Bluetooth Frequency Deviation vs. Time display. (The  $f_n - f_0$  indicates the drift of the frequency offset calculated in each payload interval from the offset calculated in the preamble.)

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO:INTERVAL?
```

## Arguments

None

## Returns

<NR1> the interval at which the maximum drift ( $f_n - f_0$ ) occurred.

## Examples

READ:BLUETOOTH:FREQUENCY:OFFSET:DRIFT:FNFZERO:INTERVAL? might return 16, indicating that at the 16th payload interval, the maximum drift occurred.  $f(16) - f(0)$  is the maximum drift among all  $f(n) - f(0)$ .

## READ:BLUETOOTH:FREQUENCY:OFFSET:MAX? (Query Only)

Returns the maximum frequency offset in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time, CF Offset

## Group

Read commands

## Syntax

`READ:BLUetooth:FREQuency:OFFSet:MAX?`

## Arguments

None

## Returns

<NR3> the maximum frequency offset value.

## Examples

`READ:BLUetooth:FREQuency:OFFSet:MAX?` might return -189.4632263184, the frequency offset value.

## **READ:BLUetooth:FREQuency:OFFSet:MAX:INTERval? (Query Only)**

Returns the maximum frequency offset interval in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

`READ:BLUetooth:FREQuency:OFFSet:MAX:INTERval?`

## Arguments

None

## Returns

<NR1> the maximum frequency offset interval.

## Examples

`READ:BLUetooth:FREQuency:OFFSet:MAX:INTERval?` might return 271 indicating that the maximum drift  $f(n) - f(0)$  occurred at the 271st payload interval.

## **READ:BLUetooth:FREQuency:OFFSet:PREAmble? (Query Only)**

Returns the offset calculated in the preamble region in the Frequency Deviation vs. Time and the Center Frequency Offset and Drift displays.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

```
READ:BLUetooth:FREQuency:OFFSet:PREAmble?
```

## Arguments

None

## Returns

<NR3> the offset calculated in the preamble region.

## Examples

READ:BLUetooth:FREQuency:OFFSet:PREAmble? might return 106.4204711914, indicating the offset calculated in the preamble region.

## READ:BLUetooth:MODUlation:CHARacteristics:AVERage:F? (Query Only)

Returns the average frequency value when the selected test pattern is Other in the Bluetooth Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Read commands

## Syntax

```
READ:BLUetooth:MODUlation:CHARacteristics:AVERage:F?
```

## Arguments

None

## Returns

<NRf> the average frequency.

## Examples

READ:BLUetooth:MODUlation:CHARacteristics:AVERage:F? might return 140.8309531250E+3, the average frequency value.

## **READ:BLUetooth:MODUlation:CHARacteristics:AVErage:F2F1ratio? (Query Only)**

Returns the average ratio value for the high and low frequency deviation in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Read commands

### **Syntax**

`READ:BLUetooth:MODUlation:CHARacteristics:AVErage:F2F1ratio?`

### **Arguments**

None

### **Returns**

<NR1> the average ratio value.

### **Examples**

`READ:BLUetooth:MODUlation:CHARacteristics:AVErage:F2F1ratio?` might return `????`, the average ratio value.

## **READ:BLUetooth:MODUlation:CHARacteristics:AVErage:FONE? (Query Only)**

Returns the average frequency deviation for the low deviation pattern in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Read commands

### **Syntax**

`READ:BLUetooth:MODUlation:CHARacteristics:AVErage:FONE?`

### **Arguments**

None

### **Examples**

`READ:BLUetooth:MODUlation:CHARacteristics:AVErage:FONE?` might return `139.8309531250E+3`, the average frequency deviation for the low deviation pattern.



## **READ:BLUETOOTH:MODULATION:CHARACTERISTICS:AVERAGE:FTWO? (Query Only)**

Returns the average frequency deviation for the high deviation pattern in the Bluetooth Frequency Deviation vs. Time display.

### **Conditions**

Measurement views: Frequency Deviation vs. Time

### **Group**

Read commands

### **Syntax**

`READ:BLUETOOTH:MODULATION:CHARACTERISTICS:AVERAGE:FTWO?`

### **Arguments**

None

### **Examples**

`READ:BLUETOOTH:MODULATION:CHARACTERISTICS:AVERAGE:FTWO?` might return `139.8309531250E+3`, the average frequency deviation for the high deviation pattern.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:COUNT? (Query Only)**

Returns the number of packets analyzed to obtain average drift results from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:COUNT?`

### **Arguments**

None

### **Returns**

<NR1> the number of packets analyzed.

### **Examples**

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:COUNT?` might return `8`, the number of packets analyzed.

## **READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero? (Query Only)**

Returns the drift f1–f0 from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero?
```

### **Arguments**

None

### **Returns**

<NRf> the drift f1–f0 value.

### **Examples**

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:F1FZERO? might return 2.846E+3 indicating that the drift f1–f0 value is 2.846 kHz.

## **READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero:STATus? (Query Only)**

Returns the Pass/Fail drift f1–f0 result status in the packet from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:F1FZero: STATus?
```

### **Arguments**

None

### **Returns**

<string> the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:F1FZERO:STATUS?` might return PASS, FAIL, or NA indicating whether the drift f1–f0 is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5? (Query Only)**

Returns the maximum drift of the fn–fn-5 value from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5?`

### Arguments

None

### Returns

<NRf> the fn–fn-5 value.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5?` might return 2.846E+3 indicating that the maximum drift fn–fn-5 value in the preamble region is -2.846 kHz.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5:STATUS? (Query Only)**

Returns the Pass/Fail status of the maximum fn–fn-5 drift in the packet from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5:STATUS?`

### Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFN5:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum drift  $f_n-f_{n-5}$  is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZero? (Query Only)**

Returns the maximum drift  $f_n-f_0$  from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZero?`

## Arguments

None

## Returns

<NRf> the  $f_n-f_0$  value.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZERO?` might return 4.846E+3 indicating that the drift  $f_n-f_0$  value in the preamble region is 4.846 kHz.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZero:STATUS? (Query Only)**

Returns the Pass/Fail result status of the  $f_n-f_0$  maximum drift in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZero: STATUS?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:FNFZERO:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum drift  $f_n - f_0$  is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:CODING:SCHEME (Query Only)**

Returns the offset drift coding scheme value (LE Coded) in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:CODING:SCHEME?`

## Arguments

None

## Returns

<NR1> 2 or 8

Where 2 is data at 500 Kb/s and 8 is 125 Kb/s

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:CODING:SCHEME` might return "8", indicating the offset drift coding scheme value is 8.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE (Query Only)**

Returns the Drift value  $f_0 - f_3$  for LE Coded in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE?`

## Arguments

None

## Returns

<NRf> Maximum Frequency value in kHz

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE` might return 207.2 kHz, indicating the f0-f3 drift value is 207.2 kHz.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS (Query Only)**

Returns the Drift value Status of f0-f3 for LE Coded in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS?`

## Arguments

None

## Returns

<String> PASS , FAIL or NA

PASS indicates that the test passed

FAIL indicates that the test failed

NA indicates that the test is not Applicable

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F0FTHREE:STATUS` might return Pass, indicating the Drift value Status of f0-f3 was Pass.

## **READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour (Query Only)**

Returns the Drift value of f1-f4 for LE Coded in Bluetooth Summary display.

### **Conditions**

Measurement view: Bluetooth Summary display

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour?
```

### **Arguments**

None

### **Returns**

<NRf> Maximum Frequency value in kHz

### **Examples**

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F1FFOUR might return -10.22 kHz, indicating the f1-f4 drift value is -10.22 kHz.

## **READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour:STATus (Query Only)**

Returns the Drift value Status of f1-f4 for LE Coded in Bluetooth Summary display.

### **Conditions**

Measurement view: Bluetooth Summary display

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:FREQuency:OFFSet:DRIFt:LECoded:F1FFour:STATus?
```

### **Arguments**

None

### **Returns**

<String> PASS , FAIL or NA

PASS indicates that the test passed

FAIL indicates that the test failed

NA indicates that the test is not Applicable

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:F1FFOUR:STATUS` might return Pass, indicating the Drift value Status of f1-f4 was Pass.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3 (Query Only)**

Returns the Drift value of fn-f(n-3) for LE Coded in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3?`

### Arguments

None

### Returns

<NRf> Maximum Frequency value in kHz

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3` might return -10.42 kHz, indicating the fn-f(n-3) drift value is -10.42 kHz.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3:STATUS (Query Only)**

Returns the Drift value Status of fn-f(n-3) for LE Coded in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3:STATUS?`



## Arguments

None

## Returns

<String> PASS , FAIL or NA

PASS indicates that the test passed

FAIL indicates that the test failed

NA indicates that the test is not Applicable

## Examples

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:DRIFT:LECODED:FNFN3:STATUS might return Pass, indicating the Drift value Status of fn-f(n-3) was Pass.

## READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX? (Query Only)

Returns the maximum frequency offset present in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX?

## Arguments

None

## Returns

<NRf> the maximum frequency offset value in the packet.

## Examples

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX? might return 7.446E+3 indicating that the maximum frequency offset value in the packet is 7.446 kHz.

## READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX:STATUS? (Query Only)

Returns the Pass/Fail result status of the frequency offset maximum in the packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX:STATUS?`

## Arguments

None

## Returns

`<string>` the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:MAX:STATUS?` might return PASS, FAIL, or NA indicating whether the maximum frequency offset in the packet is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble? (Query Only)**

Returns the frequency offset value in the preamble region from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble?`

## Arguments

None

## Returns

`<NRf>` the frequency offset value in the preamble region.

## Examples

`READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble?` might return 3.546E+3 indicating that the frequency offset value in the preamble region is 3.546 kHz.

## **READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble:STATus? (Query Only)**

Returns the Pass/Fail result status of the frequency offset in the preamble region.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble:STATUS?
```

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

READ:BLUETOOTH:SUMMARY:FREQUENCY:OFFSET:PREAmble:STATUS? might return PASS, FAIL, or NA indicating whether the frequency offset in the preamble region is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:F:MAX:PERCENTAGE:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for  $\Delta F_{avg}$  from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:F:MAX:PERCENTAGE:COUNT?
```

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:F:MAX:PERCENTAGE:COUNT? might return 3 indicating that three packets were analyzed.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FONE:MAX:PERCENTAGE:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for the  $\Delta F1$ avg from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FONE:MAX:PERCENTAGE:COUNT?

### **Arguments**

None

### **Returns**

<NR1> the number of packets analyzed.

### **Examples**

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FONE:MAX:PERCENTAGE:COUNT? might return 6 indicating that six packets were analyzed.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE? (Query Only)**

Returns the percentage of the  $\Delta F2$ max% values that are above the specified limit.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE?

### **Arguments**

None

## Returns

<NRf> the number of packets analyzed.

## Examples

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE ? might return 99.2 indicating that 99.2% of the deltaF2 maximum values are above the specified limit.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for delta F2 average from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE :COUNT?

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE :COUNT? might return 6 indicating that six packets were analyzed.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:STATUS? (Query Only)**

Returns the Pass/Fail result status of the  $\Delta F2Max\%$  field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:STATUS?`

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:DELTA:FTWO:MAX:PERCENTAGE:STATUS?` might return PASS, FAIL, or NA indicating the Pass/Fail status of the  $\Delta F2Max\%$  field in the Bluetooth summary.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1ratio? (Query Only)**

Returns the ratio of the  $\Delta F2avg$  to  $\Delta F1avg$  from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1ratio?`

## Arguments

None

## Returns

<NRf> the value of the  $\Delta F1avg$  to  $\Delta F2avg$  ratio.

## Examples

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1RATIO?` might return 225.12E+3 indicating the average ratio is 22.5.12 kHz.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1ratio:STATUS? (Query Only)**

Returns the Pass/Fail results status of the  $\Delta F2avg/\Delta F1avg$  field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F2F1ratio:STATus?
```

## Arguments

None

## Returns

<string> the Pass/Fail result status.

## Examples

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F2F1RATIO:STATUS? might return PASS, FAIL, or NA indicating the Pass/Fail result status of the  $\Delta F_{2avg}/\Delta F_{1avg}$  field.

## READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F? (Query Only)

Returns the  $\Delta F_{avg}$  value from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUEtooth:SUMMary:MODUlation:CHARacteristics:AVERage:F?
```

## Arguments

None

## Returns

<NRf> the  $\Delta F_{avg}$  value.

## Examples

READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:F? might return 2.8723E+3 indicating the  $\Delta F_{avg}$  is 2.8723 kHz.

## **READ:BLUetooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE? (Query Only)**

Returns the delta F1 average value from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

`READ:BLUetooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE?`

### **Arguments**

None

### **Returns**

<NRf> the delta F1 value.

### **Examples**

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FONE?` might return `155.3672E+3` indicating the delta F1 average is 155.3672 kHz.

## **READ:BLUetooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE:STATus? (Query Only)**

Returns the Pass/Fail results status of the delta F1 average from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

`READ:BLUetooth:SUMMary:MODUlation:CHARacteristics:AVErage:FONE:STATus?`

### **Arguments**

None

### **Returns**

<string> the Pass/Fail results status.



---

## Examples

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FONE:STATUS?` might return PASS, FAIL, or NA indicating whether the delta F1 average is within the limit.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FTWO? (Query Only)**

Returns the delta F2 average value from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FTWO?`

### Arguments

None

### Returns

<NRf> the delta F1 value.

## Examples

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:AVERAGE:FTWO?` might return 225.12E+3 indicating that the delta F2 value is 225.12 kHz.

## **READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:INDEX (Query Only)**

Returns the Modulation index value in Bluetooth Summary display.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Read commands

### Syntax

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:INDEX?`

### Arguments

None

## Returns

<String>

## Examples

`READ:BLUETOOTH:SUMMARY:MODULATION:CHARACTERISTICS:INDEX` might return "Stable", indicating the status of modulation index is Stable.

## **READ:BLUetooth:SUMMary:OUTPut:POWER:AVerage? (Query Only)**

Returns the average output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUetooth:SUMMary:OUTPut:POWER:AVerage?`

## Arguments

None

## Returns

<NRf> the average output power value in dBm.

## Examples

`READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:AVERAGE?` might return -29.1 indicating that the average output power is -29.1 dBm.

## **READ:BLUetooth:SUMMary:OUTPut:POWER:AVerage:STATus? (Query Only)**

Returns the Pass/Fail results status of the average output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUetooth:SUMMary:OUTPut:POWER:AVerage:STATus?`

## Arguments

None

## Returns

<string> the Pass/Fail results status.

## Examples

`READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:AVERAGE:STATUS?` might return PASS, FAIL, or NA indicating whether the average output power is within the specified limit.

## **READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for the calculation of the output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:COUNT?`

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

`READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:COUNT?` might return 6 indicating that six packets were analyzed.

## **READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:COUNT? (Query Only)**

Returns the number of packets analyzed out of 10 packets for calculation of the output power for EDR from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:COUNT?

## Arguments

None

## Returns

<NR1> the number of packets analyzed.

## Examples

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:COUNT? might return 6 indicating that six packets were analyzed.

## READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:DPSK? (Query Only)

Returns the power in the DPSK portion of the EDR burst from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:DPSK?

## Arguments

None

## Returns

<NRf> the power in the DPSK portion of the EDR burst.

## Examples

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:DPSK? might return -29.2 indicating that the average output power is -29.2 dBm.

## READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:GFSK? (Query Only)

Returns the power in the GFSK portion of the EDR burst from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:GPSK?
```

## Arguments

None

## Returns

<NRf> the power in the GFSK portion of the EDR burst.

## Examples

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:EDR:GPSK? might return -29.1 indicating that the average output power is -29.1 dBm.

## READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK? (Query Only)

Returns the peak output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK?
```

## Arguments

None

## Returns

<NRf> the peak output power value in dBm.

## Examples

READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK? might return -29.3 indicating that the peak output power is -29.3 dBm.

## READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK:STATUS? (Query Only)

Returns the Pass/Fail results status of the peak output power from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUEtooth:SUMMary:OUTPut:POWEr:PEAK:STATus?`

## Arguments

None

## Returns

`<string>` the Pass/Fail result status.

## Examples

`READ:BLUETOOTH:SUMMARY:OUTPUT:POWER:PEAK:STATUS?` might return PASS, FAIL, or NA indicating whether the peak output power is within the specified limit.

## **READ:BLUEtooth:SUMMary:PACKet:BDR:CRC? (Query Only)**

Returns the Basic Rate value of the CRC field from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUEtooth:SUMMary:PACKet:BDR:CRC?`

## Arguments

None

## Returns

`<string>` the value of the CRC field.

## Examples

`FETCh:BLUEtooth:SUMMary:PACKet:BDR:CRC?` Might return "0x9DB0" indicating the value of the CRC field in hexadecimal.

---

## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:ARQN? (Query Only)**

Returns the value of ARQN field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:ARQN?

### **Arguments**

None

### **Returns**

<NR2> the value of the ARQN field.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:ARQN? might return 0.000000 indicating the value of the flow field.

## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:FLOW? (Query Only)**

Returns the value of Flow field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:FLOW?

### **Arguments**

None

### **Returns**

<NR2> the flow value.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:FLOW? might return 1.000000 indicating the value of the flow field.

## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:HEC? (Query Only)**

Returns the value of HEC field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:HEC?

### **Arguments**

None

### **Returns**

<string> the value of the HEC field.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:HEC? might return "00110010" indicating the value of the HEC field in binary.

## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:LT:ADDRess? (Query Only)**

Returns the LT address from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:LT:ADDRess?

### **Arguments**

None

### **Returns**

<string> the LT address.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:LT:ADDR? might return "101" indicating the LT address in binary.



## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:SEQN? (Query Only)**

Returns the value of SEQN field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:SEQN?
```

### **Arguments**

None

### **Returns**

<NR2> the value of the SEQN field.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:SEQN? might return 1.000000 indicating the value of the flow field.

## **READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:TYPE? (Query Only)**

Returns the value of Type field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:PACKet:BDR:HEADer:TYPE?
```

### **Arguments**

None

### **Returns**

<string> the value of the type field.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:HEADer:TYPE? might return "00001111" indicating the type field in binary.

## **READ:BLUetooth:SUMMary:PACKet:BDR:PAYLoad:LENgth? (Query Only)**

Returns the Basic Rate payload length from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:PAYLoad:LENgth?

### **Arguments**

None

### **Returns**

<string> the value of the payload length.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:PAYLoad:LENgth? might return "0101010011" indicating the payload length value in binary.

## **READ:BLUetooth:SUMMary:PACKet:BDR:PREAmble? (Query Only)**

Returns the preamble of the Basic Rate packet from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BDR:PREAmble?

### **Arguments**

None

### **Returns**

<string> the preamble of the packet.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:PREAmble? might return "0101" indicating the preamble is 0101 in binary.

## **READ:BLUetooth:SUMMary:PACKet:BDR:SYNC:WORD? (Query Only)**

Returns the sync word from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:PACKet:BDR:SYNC:WORD?
```

### **Arguments**

None

### **Returns**

<string> the sync word.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:SYNC:WORD? might return "0x4F36F2CEE85390CB" indicating the sync word in hexadecimal format.

## **READ:BLUetooth:SUMMary:PACKet:BDR:TYPE? (Query Only)**

Returns the packet type of the Basic Rate signal from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUetooth:SUMMary:PACKet:BDR:TYPE?
```

### **Arguments**

None

### **Returns**

<string> the packet type.

### **Examples**

FETCh:BLUetooth:SUMMary:PACKet:BDR:TYPE? might return "DH5" indicating that the DH5 packet is being analyzed.

## **READ:BLUetooth:SUMMary:PACKet:BLE:ACCEss:ADDRess? (Query Only)**

Returns the 32-bit Access Address from the Bluetooth Summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BLE:ACCEss:ADDRess?

### **Arguments**

None

### **Returns**

<string> the 32-bit Access Address.

### **Examples**

READ:BLUETOOTH:SUMMARY:PACKET:BLE:ACCESS:ADDRESS? might return "0x71764129" indicating the Access Address in hexadecimal.

## **READ:BLUetooth:SUMMary:PACKet:BLE:CRC? (Query Only)**

Returns the value of the CRC field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

READ:BLUetooth:SUMMary:PACKet:BLE:CRC?

### **Arguments**

None

### **Returns**

<string> the value of the CRC field.

### **Examples**

READ:BLUETOOTH:SUMMARY:PACKET:BLE:CRC? might return "0x9DB0" indicating value CRC field in hexadecimal.

## **READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:CHANNEL:SELECT (Query Only)**

Returns the channel selected under decoded packet information section in Bluetooth Summary display.

### **Conditions**

Measurement view: Bluetooth Summary display

### **Group**

Read commands

### **Syntax**

```
READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:CHANNEL:SELECT?
```

### **Arguments**

None

### **Returns**

<String> 0 or 1 or -1 or “-”

0 and 1 represents valid

-1 represents invalid

“-” represents NIL

### **Examples**

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:CHANNEL:SELECT might return 1, indicating the selected channel is valid.

## **READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LENGTH? (Query Only)**

Returns the value of the PDU length field from the Bluetooth summary.

### **Conditions**

Measurement views: Bluetooth summary

### **Group**

Read commands

### **Syntax**

```
READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LENGTH?
```

### **Arguments**

None

## Returns

<string> the value of the length field.

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LENGTH? might return "100101" indicating value length field in binary.

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LLID (Query Only)

Returns the LLID value (2 bits) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LLID?

## Arguments

None

## Returns

<String> Either 00,01,10,11

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:LLID might return 01, indicating the value of LLID is "01".

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:MD (Query Only)

Returns the MD value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:MD?

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:MD might return 1, indicating the value of MD is "1".

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:NESN (Query Only)

Returns the NESN value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:NESN?

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:NESN might return 1, indicating the value of NESN is "1".

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:RX:ADDRESS? (Query Only)

Returns the PDU Rx address from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:RX:ADDRESS?

## Arguments

None

## Returns

<string> the Rx address.

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:RX:ADDRESS? might return 0.0000000 indicating Rx Address.

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:SN (Query Only)

Returns the SN value (1 bit) under decoded packet information when the packet type is "OTHERS" for Low Energy in Bluetooth Summary display.

## Conditions

Measurement view: Bluetooth Summary display

## Group

Read commands

## Syntax

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:SN?

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:SN might return 1, indicating the value of SN is "1".

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TX:ADDRESS? (Query Only)

Returns the PDU Tx address from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary



## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TX:ADDRESS?
```

## Arguments

None

## Returns

<string> the Tx address.

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TX:ADDRESS? might return 0.00000000 indicating Tx Address.

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TYPE? (Query Only)

Returns the PDU packet type from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

```
READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TYPE?
```

## Arguments

None

## Returns

<string> the PDU type.

## Examples

READ:BLUETOOTH:SUMMARY:PACKET:BLE:PDU:HEADER:TYPE? might return "0010" indicating the PDU type in binary.

## READ:BLUETOOTH:SUMMARY:PACKET:BLE:PREAMBLE? (Query Only)

Returns the preamble of the BLE packet from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:PACKET:BLE:PREAmble?`

## Arguments

None

## Returns

`<string>` the preamble of the packet.

## Examples

`READ:BLUETOOTH:SUMMARY:PACKET:BLE:PREAmble?` might return "01010101" indicating the preamble is 01010101 in binary.

## **READ:BLUETOOTH:SUMMARY:PACKET:BLE:TYPE? (Query Only)**

Returns the packet type from the Bluetooth summary.

## Conditions

Measurement views: Bluetooth summary

## Group

Read commands

## Syntax

`READ:BLUETOOTH:SUMMARY:PACKET:BLE:TYPE?`

## Arguments

None

## Returns

`<string>` the BLE packet type.

## Examples

`READ:BLUETOOTH:SUMMARY:PACKET:BLE:TYPE?` might return "BLE\_TEST" indicating the BLE test signal is being analyzed.

## READ:CCDF? (Query Only)

Returns the CCDF measurement results.

### Conditions

Measurement views: CCDF

### Group

Read commands

### Syntax

```
READ:CCDF?
```

### Arguments

None

### Returns

```
<avg_ampl>,<avg_ccdf>,<crest_factor>,<ampl_10>,<ampl_1>,  
<ampl_p1>,<ampl_p01>,<ampl_p001>,<ampl_p0001>
```

Where

<avg\_ampl> is the average amplitude in dBm.

The unit can be changed by the [SENSE]:POWER:UNITs command.

<avg\_ccdf> is the average CCDF in percent.

<crest\_factor> is the crest factor in dB.

<ampl\_10> is the amplitude at CCDF of 10% in dB.

<ampl\_1> is the amplitude at CCDF of 1% in dB.

<ampl\_p1> is the amplitude at CCDF of 0.1% in dB.

<ampl\_p01> is the amplitude at CCDF of 0.01% in dB.

<ampl\_p001> is the amplitude at CCDF of 0.001% in dB.

<ampl\_p0001> is the amplitude at CCDF of 0.0001% in dB.

### Examples

READ:CCDF? might return -33.35,35.8,9.75,3.88,7.07,8.50,9.25,9.72,9.74, indicating

(average amplitude) = -33.35 dBm,

(average CCDF) = 35.8%,

(crest factor) = 9.75 dB,

(amplitude at CCDF of 10%) = 3.88 dB,

(amplitude at CCDF of 1%) = 7.07 dB,

(amplitude at CCDF of 0.1%) = 8.50 dB,

(amplitude at CCDF of 0.01%) = 9.25 dB,

(amplitude at CCDF of 0.001%) = 9.72 dB, and

(amplitude at CCDF of 0.0001%) = 9.74 dB.

## READ:CCDF:{FIRSt|SECond|THIRd}:X? (Query Only)

Returns the horizontal values of the specified trace in the CCDF measurement. The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.



**Note:** This query is invalid when [SENSe]:CCDF:TIME:TYPE is set to CONTinuous or TOTal.

### Conditions

Measurement views: CCDF

### Group

Read commands

### Syntax

```
READ:CCDF:{FIRSt|SECond|THIRd}:X?
```

### Related Commands

```
READ:CCDF:{FIRSt|SECond|THIRd}[:Y]?
```

### Arguments

None

### Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the horizontal value (dB) of the CCDF graph at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

READ:CCDF:FIRSt:X might return #41024xxxx... (1024-byte data) for the horizontal values of Trace 1.

## READ:CCDF:{FIRSt|SECond|THIRd}:XY? (Query Only)

Returns the horizontal and vertical value pairs of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.



**Note:** This query is invalid when [SENSE] :CCDF :TIME :TYPE is set to CONTinuous or TOTal.

## Conditions

Measurement views: CCDF

## Group

Read commands

## Syntax

```
READ:CCDF:{FIRSt|SECond|THIRd}:XY?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the horizontal value (dB) and vertical value (%) pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:CCDF:FIRSt:XY? might return #41024xxxx... (1024-byte data) for the horizontal and vertical value pairs of Trace 1.

## READ:CCDF:{FIRSt|SECond|THIRd}[:Y]? (Query Only)

Returns the vertical values of the specified trace in the CCDF measurement.

The mnemonics FIRSt, SECond, and THIRd represent Trace 1, Trace 2, and Gaussian reference curve, respectively.



**Note:** This query is invalid when [SENSE] :CCDF :TIME :TYPE is set to CONTinuous or TOTal.

## Conditions

Measurement views: CCDF

## Group

Read commands

## Syntax

```
READ:CCDF:{FIRSt|SECond|THIRd}[:Y]?
```

## Arguments

None

## Returns

#<num\_digit><num\_byte><y(1)><y(2)>...<y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the vertical value (%) of the CCDF graph at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:CCDF:FIRST:Y might return #41024xxxx... (1024-byte data) for the vertical values of Trace 1.

## READ:CONSte[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user selected center frequency of the analyzer.

The parameter <x> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Read commands

## Syntax

READ:CONSte[:MEASview<y>]:FERRor?

## Related Commands

[READ:EVM\[:MEASview<y>\]:FERRor?](#)

## Arguments

None

## Returns

<freq\_error>::<NRf> which is the frequency error in Hz.

## Examples

`READ:CONStE:MEASview2:FERRor?` might return `-10.7E+3`, which is a frequency error of `-10.7 kHz` for measview 2 of constellation.

## READ:CONStE[:MEASview<y>]:RESults? (Query only)

Returns the constellation measurement results of EVM RMS, peak, and location displayed on the bottom of the screen.

The parameter `<x>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Read commands

## Syntax

`READ:CONStE[:MEASview<y>]:RESults?`

## Arguments

None

## Returns

For modulation types 2|4|8|16FSK or C4FM:

`FSK_deviation_Avg_Leftmost`, `FSK_deviation_Avg_Rightmost`

Where

`FSK_deviation_Avg_Leftmost` is the average FSK deviation of the left-most symbol in Hz.

`FSK_deviation_Avg_Rightmost` is the average FSK deviation of the right-most symbol in Hz.

For all other valid modulation types:

`<EVM_RMS>`, `<EVM_peak>`, `<location>`

Where

`<EVM_RMS>::<NRf>` is the RMS EVM in percent (%).

`<EVM_peak>::<NRf>` is the peak EVM in percent (%).

`<location>::<NRf>` is the peak location in symbol number.

The time unit can be changed by the `[SENSE]:DDEMod:TIME:UNITs` command.

## Examples

`READ:CONSTE:MEASview2:RESULTS?` might return 2.841,3.227,68.000, indicating that the RMS EVM is 2.841% and the peak EVM is 3.227% at symbol #68 for measview 2 of constellation.

## **READ:CONStE[:MEASview<y>]:TRACe? (Query only)**

Returns the constellation trace data.

The parameter <x> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Read commands

## Syntax

`READ:CONStE[:MEASview<y>]:TRACe?`

## Arguments

None

## Returns

`#<num_digit><num_byte><I(1)><Q(1)><I(2)><Q(2)>...<I(n)><Q(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<I(n)>` and `<Q(n)>` are the normalized I- and Q-coordinate values at the  $n^{\text{th}}$  data point. 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:CONSTE:MEASview2:TRACE?` might return #43848xxxx... (3848-byte data) for the constellation trace data for measview 2 of constellation.

## **READ:DDEMod[:MEASview<y>]:STABle? (Query only)**

Returns the symbol table data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).



## Conditions

Measurement views: Symbol table

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Symbol table).

## Group

Read commands

## Syntax

```
READ:DDEMod[:MEASview<y>]:STABLE?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the symbol table data for the point n, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:DDEMod:MEASview1:STABLE? might return #3512xxx... (512-byte data) for the symbol Table measurement view1.

## READ:DIQVtime[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the Demod I&Q versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Read commands

## Syntax

```
READ:DIQVtime:[MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`READ:DIQVtime:MEASview1:FERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz measurement view 1.

## **READ:DIQVtime[:MEASview<y>]:I? (Query only)**

Returns the I versus Time trace data.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Read commands

## Syntax

`READ:DIQVtime[:MEASview<y>]:I?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data (1)><data (2)>...<data (n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data (n)>` is the I level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:DIQVtime:MEASview1:I?` might return `#3160xxx...` (160-byte data) for the I versus Time trace measurement view 1.

## READ:DIQVtime[:MEASview<y>]:Q? (Query only)

Returns the Q versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Read commands

### Syntax

```
READ:DIQVtime[:MEASview<y>]:Q?
```

### Arguments

None

### Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the Q level in volts at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

READ:DIQVtime:MEASview1:Q? might return #3160xxxx... (160-byte data) for the Q versus Time trace measurement view 1.

## READ:DPX:DDENsity? (Query Only)

Returns the calculated density for the selected region in the DPXogram measurement.

### Conditions

Measurement views: DPX, Spectrum

### Group

Read commands

### Syntax

```
READ:DPX:DDENsity?
```

## Arguments

None

## Returns

<NR1> value.

## Examples

READ:DPX:DDENSITY? might return a DPX signal density value of 24.4802413393.

## READ:DPX:DGRam:LINE:COUNT? (Query Only)

Returns the number of lines in the DPXogram measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

READ:DPX:DGRam:LINE:COUNT?

## Arguments

None

## Returns

<NR1> number of lines in the measurement.

## Examples

READ:DPX:DGRAM:LINE:COUNT? might return 40 indicating that there were 40 lines in the measurement.

## READ:DPX:DGRam:TIME[:SCALE]:OFFSet? (Query Only)

Returns the time scale offset in the DPXogram measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

READ:DPX:DGRam:TIME[:SCALE]:OFFSet?

## Arguments

None

## Returns

<string> The time offset in the DPXogram measurement in seconds.

## Examples

READ:DPX:DGRAM:TIME[:SCALE]:OFFSET? might return "0.000 s" indicating that the time scale offset was 0.000 s.

## READ:DPX:RESults:TRACe<x>? (Query Only)

Acquires a waveform and then returns waveform data of the specified trace <x> in the DPX spectrum measurement, where x is 1 to 6. The traces 1–4 are in the standard form. Trace 5 is the bitmap trace and its data is returned in a binary block. Trace 6 is the DPXogram trace on DPX spectrum plots. Trace 7 is the Ogram line in the DPXogram display.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

READ:DPX:RESults:TRACe<x>?

## Arguments

<NR1>

## Returns

For traces 1 to 4 or 6:

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the amplitude of the trace sample at the nth data point.

<data(n)> is in the 4-byte little endian floating-point format specified in IEEE 488.2.

The format is a binary block of 32 bit floating point values. Each value ranges from 0 to 1.0, where 1.0 represents 100%.

## Examples

READ:DPX:RESULTS:TRACE1? might return #42004xxxx... (2004-byte of data) for the waveform data of trace one (1).

## **READ:DPX:TRACe:AVERage? (Query Only)**

Acquires a waveform and then returns waveform data of the average trace in the DPX, Spectrum measurement.

### **Conditions**

Measurement views: DPX, Spectrum

### **Group**

Read commands

### **Syntax**

READ:DPX:TRACe:AVERage?

### **Arguments**

None

### **Returns**

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the waveform data of the average trace for the point n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

### **Examples**

READ:DPX:TRACE:AVERAGE? might return #42004xxxx... (2004-byte data) for the waveform data of the average trace.

## **READ:DPX:TRACe:BITMap? (Query Only)**

Acquires a waveform and then returns trace waveform data of the bitmap trace in the DPX, Spectrum measurement.

### **Conditions**

Measurement views: DPX, Spectrum

### **Group**

Read commands

### **Syntax**

READ:DPX:TRACe:BITMap?

### **Arguments**

None

## Returns

The format is a binary block of 32 bit floating point values. Each value ranges from 0 to 1.0, where 1.0 represents 100%.

## Examples

`READ:DPX:TRACE:BITMAP?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the bitmap trace.

## READ:DPX:TRACe:MATH? (Query Only)

Returns waveform data of the math trace in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

`READ:DPX:TRACe:MATH?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the waveform data of the math trace for the point `n` in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:DPX:TRACE:MATH?` might return `#42004xxxx...` (2004-byte data) for the waveform data of the math trace.

## READ:DPX:TRACe:MAXimum? (Query Only)

Returns waveform data of the maximum trace in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

READ:DPX:TRACe:MAXimum?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the waveform data of the maximum trace for the point n in dBm, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSE]:POWER:UNITs command.

## Examples

READ:DPX:TRACE:MAXIMUM? might return #42004xxxx... (2004-byte data) for the waveform data of the maximum trace.

## READ:DPX:TRACe:MINimum? (Query Only)

Returns waveform data of the minimum trace in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Read commands

## Syntax

READ:DPX:TRACe:MINimum?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.



<num\_byte> is the number of bytes of data that follow.

<data (n) > is the waveform data of the minimum trace for the point n in dBm.

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:DPX:TRACE:MINIMUM? might return #42004xxxx... (2004-byte data) for the waveform data of the minimum trace.

## READ:EDIagram[:MEASview<y>]:FDEVIation? (Query only)

Returns the frequency deviation versus Time trace data with the X values.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Group

Read commands

## Syntax

READ:EDIagram[:MEASview<y>]:FDEVIation?

## Returns

#<num\_digit><num\_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> is frequency deviation in Hz and time (symbols) pair at the n<sup>th</sup> data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:EDIagram:MEASview2:FDEVIation? might return #3160xxxx... (160-byte data) for the frequency deviation versus Time trace for measview2 of the Eye diagram.

## READ:EDIagram[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the eye diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Read commands

## Syntax

```
READ:EDIagram[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

READ:EDIagram:MEASview2:FERRor? might return -10.7E+3, indicating the frequency error is -10.7 kHz for measview2 of the Eye diagram.

## READ:EDIagram[:MEASview<y>]:I? (Query only)

Returns the I versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Read commands

## Syntax

```
READ:EDIagram[:MEASview<y>]:I?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

$\langle Y(n) \rangle \langle X(n) \rangle$  is the I level (normalized) and time (symbols) pair at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:EDIagram:MEASview2:I?` might return #3160xxxx... (160-byte data) for the I versus Time trace for measview2 of the Eye diagram.

## READ:EDIagram[:MEASview<y>]:Q? (Query only)

Returns the Q versus Time trace data.

The parameter  $\langle y \rangle = 1$  to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Read commands

## Syntax

`READ:EDIagram[:MEASview<y>]:Q?`

## Arguments

None

## Returns

$\# \langle \text{num\_digit} \rangle \langle \text{num\_byte} \rangle \langle \text{data}(1) \rangle \langle \text{data}(2) \rangle \dots \langle \text{data}(n) \rangle$

Where

$\langle \text{num\_digit} \rangle$  is the number of digits in  $\langle \text{num\_byte} \rangle$ .

$\langle \text{num\_byte} \rangle$  is the number of bytes of data that follow.

$\langle Y(n) \rangle \langle X(n) \rangle$  is the Q level (normalized) and time (symbols) pair at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:EDIagram:MEASview2:Q?` might return #3160xxxx... (160-byte data) for the Q versus Time trace for measview2 of the Eye diagram.

## READ:EMC:COUNT? (Query Only)

Returns the total numbers of spots detected in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:COUNT?

## Arguments

None

## Returns

<NR1>

## Examples

READ:EMC:COUNT? might return 12, indicating that 12 spots are detected.

## READ:EMC:EMISsion:RESUltS:STATUs? (Query Only)

Returns the status of the emission results in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:EMISsion:RESUltS:STATUs?

## Arguments

None

## Returns

<String> either PASS or FAIL

## Examples

READ:EMC:EMISSION:RESULTS:STATUS? might return FAIL, indicating that the status of emission results is failed.

## READ:EMC:SPOT<x>:AMBlent:AMPLitude? (Query Only)

Returns the amplitude of the selected spot in ambient trace of EMC View.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

```
READ:EMC:SPOT<x>:AMBIent:AMPLitude?
```

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT1:AMBIent:AMPLitude? might return 52.612, indicating that amplitude of spot 1 in ambient trace is 52.61 dBuV.

## READ:EMC:SPOT<x>:AMBIent:DELTA? (Query Only)

Returns the Delta value of the selected spot in ambient trace of EMC View.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

```
READ:EMC:SPOT<x>:AMBIent:DELTA?
```

## Arguments

None

## Returns

<NRf>

## Examples

`READ:EMC:SPOT<x>:AMBIent:DELTA?` might return 3.3614, indicating that delta value of spot 2 in the ambient trace is 3.36 dBuV.

## **READ:EMC:SPOT<x>:AMBIent:PASS? (Query Only)**

Returns the status of the selected spot in ambient trace in EMC view. The default status is THRE.

<x> = the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

`READ:EMC:SPOT<x>:AMBIent:PASS?`

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

`READ:EMC:SPOT3:AMBIent:PASS?` might return THRE, indicating that the spot 3 of ambient trace is failing by crossing Threshold.

## **READ:EMC:SPOT<x>:FREQuency? (Query Only)**

Returns the frequency of the specified spot in EMC View.

<x> = <NR1> Indicates the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:FREQuency?

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT2:FREQuency? might return 366.1250000E+3, indicating the frequency of spot 2 is 366.125 kHz.

## READ:EMC:SPOT<x>:PASS? (Query Only)

Returns the overall status of the specified spot in EMC view. The default status is THRE.

<x> = <NR1> Indicates the spot number

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:PASS?

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

READ:EMC:SPOT1:PASS? might return OLIM, indicating that spot 1 is failing by crossing the Over Limit.

## READ:EMC:SPOT<x>:RANGe? (Query Only)

Returns the range of the selected spot in EMC View.

<x> = <NR1> Indicates the spot number

### Conditions

Measurement view: EMC - EMI Display

### Group

Read commands

### Syntax

READ:EMC:SPOT<x>:RANGe?

### Arguments

None

### Returns

<String> from A to T

### Examples

READ:EMC:SPOT1:RANGe? might return "B", indicating that the range of spot 1 is "B".

## READ:EMC:SPOT<x>:RECALL:AMPLitude? (Query Only)

Returns the amplitude of the specified spot in the recalled trace of EMC View.

<x> = Indicates the spot number

### Conditions

Measurement view: EMC - EMI Display

### Group

Read commands

### Syntax

READ:EMC:SPOT<x>:RECALL:AMPLitude?

### Arguments

None

### Returns

<NRf>

### Examples

READ:EMC:SPOT1:RECALL:AMPLitude? might return 23.598, indicating that amplitude of spot 1 in the recalled trace is 23.59 dBuV.



## **READ:EMC:SPOT<x>:RECALL:DELTA? (Query Only)**

Returns the Delta value of the selected spot in recalled trace of EMC View.

<x> = the spot number

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Read commands

### **Syntax**

```
READ:EMC:SPOT<x>:RECALL:DELTA?
```

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

READ:EMC:SPOT2:RECALL:DELTA? might return 5.5878, indicating that delta value of spot 2 in the recalled trace is 5.58 dBuV.

## **READ:EMC:SPOT<x>:RECALL:PASS? (Query Only)**

Returns the status of the specified spot in recalled trace in EMC view. The default status is THREE.

<x> = the spot number

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Read commands

### **Syntax**

```
READ:EMC:SPOT<x>:RECALL:PASS?
```

### **Arguments**

None

### **Returns**

<String> Either OLIM, OMAR or THREE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

READ:EMC:SPOT<x>:RECALL:PASS? might return OMAR, indicating that the spot 1 of recalled trace is failing by crossing Margin.

## READ:EMC:SPOT<x>:SCAN<y>:AMPLitude? (Query Only)

Returns the amplitude of the specified spot with respect to specified scan detector in the results table.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:SCAN<y>:AMPLitude?

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT1:SCAN1:AMPLitude? might return 20.6209, indicating that amplitude of spot 1 for scan detector 1 is 20.62 dBuV.

## READ:EMC:SPOT<x>:SCAN<y>:DELTA? (Query Only)

Returns the Delta value of the specified spot with respect to the specified scan detector in the results table.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

```
READ:EMC:SPOT<x>:SCAN<y>:DELTA?
```

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT2:SCAN1:DELTA might return 3.6209, indicating that delta value of spot 2 for scan detector 1 is 3.62 dBuV.

## READ:EMC:SPOT<x>:SCAN<y>:PASS? (Query Only)

Returns the status of the specified spot based on the specified scan detector in EMC view. The default status is THRE.

<x> = the spot number

<y> = the scan detector number (either 1 or 2)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

```
READ:EMC:SPOT<x>:SCAN<y>:PASS?
```

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

READ:EMC:SPOT2:SCAN1:PASS? might return THRE, indicating that spot 2 of scan detector 1 is failing by crossing the threshold.

## READ:EMC:SPOT<x>:SPOT<y>:AMPLitude? (Query Only)

Returns the amplitude of the specified spot with respect to specified spot detector in the results table.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:SPOT<y>:AMPLitude?

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT1:SPOT1:AMPLitude? might return 18.365, indicating that the amplitude of spot 1 for spot detector 1 is 18.36 dBuV.

## READ:EMC:SPOT<x>:SPOT<y>:DELTA? (Query Only)

Returns the Delta value of the specified spot with respect to specified spot detector in the results table.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:SPOT<y>:DELTA?

## Arguments

None

## Returns

<NRf>

## Examples

READ:EMC:SPOT2:SPOT1:DELTA? might return 2.2578, indicating that delta value of spot 2 for spot detection 1 is 2.25 dBuV.

## READ:EMC:SPOT<x>:SPOT<y>:PASS? (Query Only)

Returns the status of the specified spot based on the specified spot detector in EMC view. The default status is THRE.

<x> = the spot number

<y> = the spot detector number (either 1, 2, or 3)

## Conditions

Measurement view: EMC - EMI Display

## Group

Read commands

## Syntax

READ:EMC:SPOT<x>:SPOT<y>:PASS?

## Arguments

None

## Returns

<String> Either OLIM, OMAR or THRE

Where

OLIM = Over Limit

OMAR = Over Margin

THRE = Threshold

## Examples

READ:EMC:SPOT1:SPOT2:PASS? might return OMAR, indicating that the spot 1 of spot detector 2 is failing by crossing the Margin.

## READ:EVM[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

### Group

Read commands

### Syntax

```
READ:EVM:[MEASVIEW<y>]:FERRor?
```

### Arguments

None

### Returns

<freq\_error>::<NRf> is the frequency error in Hz.

### Examples

READ:EVM:MEASVIEW2:FERROR? might return -10.7E+3, indicating the frequency error is -10.7 kHz for measview2 of EVM.

## READ:EVM[:MEASview<y>]:PEAK? (Query only)

Returns the peak value in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8bchannels for MSO58).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

### Group

Read commands

### Syntax

```
READ:EVM[:MEASview<y>]:PEAK?
```

### Related Commands

[READ:EVM\[:MEASview<y>\]:PINdex?](#)

## Arguments

None

## Returns

`<peak>::<NRf>` is the peak EVM value in percent (%).

## Examples

`READ:EVM:MEASview2:PEAK?` might return 1.32, indicating the peak EVM value is 1.32% for measview2 of EVM.

## **READ:EVM[:MEASview<y>]:PINdex? (Query only)**

Returns the time at the EVM peak.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Read commands

## Syntax

`READ:EVM[:MEASview<y>]:PINdex?`

## Related Commands

[READ:EVM\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

`<peak_time>::<NRf>` is the time at the EVM peak in symbol number. The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`READ:EVM:MEASview2:PINDEX?` might return 68.000, indicating that the EVM peak is at symbol #68 for measview2 of EVM.

## **READ:EVM[:MEASview<y>]:RMS? (Query only)**

Returns the RMS (Root-Mean-Square) value in the EVM versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

### Group

Read commands

### Syntax

```
READ:EVM[:MEASview<y>]:RMS?
```

### Arguments

None

### Returns

<rms>::<Nrf> is the RMS EVM value in percent (%).

### Examples

READ:EVM:MEASview2:RMS? might return 0.582, indicating the RMS EVM value is 0.582% for measview2 of EVM.

## READ:EVM[:MEASview<y>]:TRACe? (Query only)

Returns the EVM versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

### Group

Read commands

### Syntax

```
READ:EVM[:MEASview<y>]:TRACe?
```

### Arguments

None



## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the EVM versus Time trace data for the point n in percent (%),

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:EVM:MEASVIEW2:TRACE? might return #42036xxx... (2036-byte data) for the EVM versus Time measview2.

## READ:FDVTime[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the Frequency deviation versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Read commands

## Syntax

READ:FDVTime:MEASview<y>:FERRor?

## Arguments

None

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

READ:FDVTIME:MEASview2:FERROR? might return -10.7E+3, indicating the frequency error is -10.7 kHz.

## READ:FDVTime[:MEASview<y>]:TRACe? (Query only)

Returns the Frequency deviation versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Read commands

## Syntax

```
READ:FDVTime:MEASview<y>:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the Frequency deviation versus Time trace data for the point n in Hz, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:FDVTime:MEASview2:TRACE? might return #3160xxx... (160-byte data) for the Frequency deviation versus Time measview 2.

## READ:{FM|PM}:FERRor? (Query only)

Returns the frequency error in the Frequency modulation and Phase modulation measurements.

## Conditions

Measurement views: Frequency and Phase modulation

## Group

Read commands

## Syntax

```
READ:{FM|PM}:FERRor?
```

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`READ:FM:FERRor?` might return `-10.7E+3`, indicating the frequency error is `-10.7` kHz.

## READ:FM:PHALf? (Query Only)

Returns the half peak-peak frequency deviation ( $Pk-Pk/2$ ) in the FM measurement.

## Conditions

Measurement views: Frequency deviation versus Time

## Group

Read commands

## Syntax

`READ:FM:PHALf?`

## Arguments

None

## Returns

`<Pk-Pk/2>::<NRf>` is the half peak-peak frequency deviation in Hz.

## Examples

`READ:FM:PHALF?` might return `628.9E+3`, indicating the half peak-peak frequency deviation is `628.9` kHz.

## READ:FM:PNEGative? (Query Only)

Returns the negative peak frequency deviation ( $-Pk$ ) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Read commands

## Syntax

`READ:FM:PNEGative?`

## Arguments

None

## Returns

<-Pk> : : <NRf> is the negative peak frequency deviation in Hz.

## Examples

READ:FM:PNEGATIVE? might return -495.6E+3, indicating the negative peak frequency deviation is -495.6 kHz.

## READ:FM:PPOSitive? (Query Only)

Returns the positive peak frequency deviation (+Pk) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Read commands

## Syntax

READ:FM:PPOSitive?

## Arguments

None

## Returns

<+Pk> : : <NRf> is the positive peak frequency deviation in Hz.

## Examples

READ:FM:PPOSITIVE? might return 763.2E+3, indicating the positive peak frequency deviation is 763.2 kHz.

## READ:FM:PTPeak? (Query Only)

Returns the peak-peak frequency deviation (Pk-Pk) in the FM measurement.

## Conditions

Measurement views: FM

## Group

Read commands

## Syntax

READ:FM:PTPeak?

## Arguments

None

## Returns

$\langle Pk-Pk \rangle : : \langle NRf \rangle$  is the peak-peak frequency deviation in Hz.

## Examples

`READ:FM:PTPEAK?` might return 1.258E+6, indicating the peak-peak frequency deviation is 1.258 MHz.

## READ:FM:RESult? (Query Only)

Returns the FM measurement results.

## Conditions

Measurement views: FM

## Group

Read commands

## Syntax

`READ:FM:RESult?`

## Arguments

None

## Returns

$\langle +Pk \rangle, \langle -Pk \rangle, \langle RMS \rangle, \langle Pk-Pk \rangle, \langle Pk-Pk/2 \rangle$

Where

$\langle +Pk \rangle : : \langle NRf \rangle$  is the positive peak frequency deviation in Hz.

$\langle -Pk \rangle : : \langle NRf \rangle$  is the negative peak frequency deviation in Hz.

$\langle RMS \rangle : : \langle NRf \rangle$  is the RMS frequency deviation in Hz.

$\langle Pk-Pk \rangle : : \langle NRf \rangle$  is the peak-peak frequency deviation in Hz.

$\langle Pk-Pk/2 \rangle : : \langle NRf \rangle$  is the half peak-peak frequency deviation in Hz.

## Examples

`READ:FM:RESULT?` might return 763.2E+3,-494.6E+3,271.2E+3,1.258E+6,628.9E+3.

## READ:FM:RMS? (Query Only)

Returns the RMS frequency deviation in the FM measurement.

## Conditions

Measurement views: FM

## Group

Read commands

## Syntax

READ:FM:RMS?

## Arguments

None

## Returns

<RMS>::<NRf> is the RMS frequency deviation in Hz.

## Examples

READ:FM:RMS? might return 271.2E+3, indicating the RMS frequency deviation is 271.2 kHz.

## READ:{FSETtling|PSETtling}:FTTime? (Query Only)

Returns the settling time from the trigger position in seconds. The settling time is measured from the trigger point (see :FETCh:FSETtling:TRIGger:TIME).

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

READ:{FSETtling|PSETtling}:FTTime?

## Arguments

None

## Returns

<value>::<NRf> is the settling time in seconds.

## Examples

READ:FSETTLING:FTTIME? might return 44.8300E-6, indicating the settling time is 44.83  $\mu$ s.

## READ:{FSETtling|PSETtling}:MASK[:PASS]? (Query Only)

Returns whether the input signal passes the mask test.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

```
READ:{FSETtling|PSETtling}:MASK[:PASS]?
```

## Arguments

None

## Returns

0 is returned if the signal fails the mask test.

1 is returned if the signal passes the mask test.

## Examples

`READ:FSETTLING:MASK:PASS?` might return 1, indicating the signal passed (did not exceed the mask limits).

## **READ:{FSETtling|PSETtling}:SETTled:FREQUency? (Query Only)**

Returns the frequency at which the signal is considered settled.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

```
READ:{FSETtling|PSETtling}:SETTled:FREQUency?
```

## Arguments

None

## Returns

`<value>::<Nrf>` is the frequency at which the signal is settled.

## Examples

`READ:FSETTLING:SETTLED:FREQUENCY?` might return 2.44838E+9, indicating the input signal frequency at the point where the signal is considered settled is 2.44838 GHz.

## **READ:{FSETtling|PSETtling}:SETTled[:PASS]? (Query Only)**

Returns whether the input signal is settled with the tolerance range.

### **Conditions**

Measurement views: Frequency and Phase Settling

### **Group**

Read commands

### **Syntax**

```
READ:{FSETtling|PSETtling}:SETTled[:PASS]?
```

### **Arguments**

None

### **Returns**

0 is returned if the signal is not settled within the tolerance range.

1 is returned if the signal is settled within the tolerance range.

### **Examples**

READ:FSETTLING:SETTLED:PASS? might return 1, indicating the signal is settled within the tolerance range.

## **READ:{FSETtling|PSETtling}:SETTled:TIME? (Query Only)**

Returns the settled time in seconds. The settled time is measured from the measurement start point.

### **Conditions**

Measurement views: Frequency and Phase Settling

### **Group**

Read commands

### **Syntax**

```
READ:{FSETtling|PSETtling}:SETTled:TIME?
```

### **Arguments**

None

### **Returns**

<value>::<Nrf> is the settled time in seconds.



## Examples

READ:FSETTLING:SETTLED:TIME? might return 299.830000E-6, indicating the settled time is 299.83  $\mu$ s.

## READ:{FSETtling|PSETtling}:SLMSd[:PASS]? (Query Only)

Returns whether the input signal is settled within the specified tolerance and the signal is settled longer than the Minimum Settled Duration ([SENSe]:{FSETtling|PSETtling}:SDURation:MINimum).

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Read commands

### Syntax

READ:{FSETtling|PSETtling}:SLMSd[:PASS]?

### Arguments

None

### Returns

0 is returned if the signal is not settled within the tolerance range and minimum settled duration..

1 is returned if the signal is settled within the tolerance range and minimum settled duration.

## Examples

READ:FSETTLING:SLMSD:PASS? might return 1, indicating the signal is settled within the tolerance range and the minimum settled duration.

## READ:{FSETtling|PSETtling}:START:TIME? (Query Only)

Returns the start time for the measurement in seconds. The start time is measured from the start of the analysis period.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Read commands

### Syntax

READ:{FSETtling|PSETtling}:START:TIME?

### Arguments

None

## Returns

<value>::<NRf> is the time in seconds when the measurement started.

## Examples

READ:FSETTLING:START:TIME? might return 251.4300E-6, indicating the measurement started 251.43  $\mu$ s after the beginning of the analysis period.

## READ:{FSETtling|PSETtling}:TIME? (Query Only)

Returns the settling time in seconds. The settling time is measured from the start time (see :FETCh:FSETtling:START:TIME).

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

READ:{FSETtling|PSETtling}:TIME?

## Arguments

None

## Returns

<value>::<NRf> is the settling time in seconds.

## Examples

READ:FSETTLING:TIME? might return 48.4000E-6, indicating the settling time is 48.4  $\mu$ s.

## READ:{FSETtling|PSETtling}:TRACe<x>:X? (Query Only)

Returns the Frequency or Phase values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

READ:{FSETtling|PSETtling}:TRACe<x>:X?

## Arguments

None

## Returns

#<num\_digit><num\_byte><y(1)><y(2)>...<y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the frequency (Hz) or phase (degrees) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:FSETTLING:TRACE1:X? might return #574232xxxx... (74232-byte data) for the frequency values of Trace 1.

## READ:{FSETtling|PSETtling}:TRACe<x>:XY? (Query Only)

Returns the time and frequency or phase value pairs of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

READ:{FSETtling|PSETtling}:TRACe<x>:XY?

## Arguments

None

## Returns

#<num\_digit><num\_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the time and frequency or phase value pair at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:FSETTLING:TRACE2:XY?` might return `#574232xxxx...` (74232-byte data) for the time and frequency pairs of the Trace 2.

## **READ:{FSETtling|PSETtling}:TRACe<x>[:Y]? (Query Only)**

Returns the frequency or phase values of the specified trace.

The parameter `<x>` = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

`READ:{FSETtling|PSETtling}:TRACe<x>[:Y]?`

## Arguments

None

## Returns

`#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<y(n)>` is the frequency or phase value at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:FSETTLING:TRACE1:Y?` might return `#575148xxxx...` (75148-byte data) for the frequency values of Trace 1.

## **READ:{FSETtling|PSETtling}:TRIGger:TIME? (Query Only)**

Returns the time when the trigger occurred in seconds.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

`READ:{FSETtling|PSETtling}:TRIGger:TIME?`

## Arguments

None

## Returns

<value>::<NRf> is the time in seconds after the measurement start point when the trigger occurred.

## Examples

READ:FSETTLING:TRIGGER:TIME? might return 2.255E-6, indicating the time when the trigger occurred 255  $\mu$ s after the start of the measurement.

## READ:{FSETtling|PSETtling}:VALue? (Query Only)

Returns the settled value in Hz for Frequency Settling and in degrees for Phase Settling.

## Conditions

Measurement views: Frequency and Phase Settling

## Group

Read commands

## Syntax

READ:{FSETtling|PSETtling}:VALue?

## Arguments

None

## Returns

<value>::<NRf> is the settling value in Hz for Frequency Settling and in degrees for Phase Settling.

## Examples

READ:FSETTLING:VALUE? might return 2.44838155E+9, indicating the settled frequency is 2.44838 GHz.

## READ:FSETtling:ERRor? (Query Only)

Returns the settled error in Hz in Frequency Settling. Only for Frequency Settling Time measurement. In Frequency Settling:

When Target Reference is set to Auto, Settled Error = 0.

## Conditions

Measurement views: Frequency Settling

## Group

Read commands

## Syntax

READ:FSEtTling:ERRor?

## Arguments

None

## Returns

<value> : : <NRf> returns the settled error in Hz.

When Target Reference is set to Auto, Settled Error = 0.

When Target Reference is set to Meas Freq:

Settled Error = Settled Frequency – (Measurement Frequency + Offset)

## Examples

READ:FSEtTLING:ERRor? might return 0, indicating the Target Reference is set to Auto.

## READ:FVTime[:MEASview<y>]? (Query Only)

Returns the Frequency versus Time trace data.

## Conditions

Measurement views: Frequency versus Time

## Group

Read commands

## Syntax

READ:FVTime[:MEASview<y>]?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the frequency drift data for the point n in Hz,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:FVTIME:MEASview1? might return #3156xxxx... (156-byte data) for the Frequency versus Time trace.

## READ:FVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Read commands

### Syntax

```
READ:FVTime[:MEASview<y>]:MAXimum?
```

### Related Commands

```
READ:FVTime[:MEASview<y>]:MAXLocation?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

<max> : : <NRf> is the maximum frequency drift in Hz.

## Examples

READ:FVTIME:MEASview1:MAXIMUM? might return 2.625E+6, indicating the maximum frequency drift is 2.625 MHz.

## READ:FVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the frequency drift is maximum.

### Conditions

Measurement views: Frequency versus Time

### Group

Read commands

### Syntax

```
READ:FVTime[:MEASview<y>]:MAXLocation?
```

## Related Commands

`READ:FVTime[:MEASview<y>]:MAXimum?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max\_time>::<Nrf> is the time in seconds at which the frequency drift is maximum.

## Examples

`READ:FVTIME:MEASview1:MAXLOCATION?` might return 25.03E-9, indicating the frequency drift is maximum at 25.03 ns.

## **READ:FVTime[:MEASview<y>]:MINimum? (Query Only)**

Returns the minimum value in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Read commands

## Syntax

`READ:FVTime[:MEASview<y>]:MINimum?`

## Related Commands

`READ:FVTime[:MEASview<y>]:MINLocation?`

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min>::<Nrf> is the minimum frequency drift in Hz.

## Examples

`READ:FVTIME:MEASview1:MINIMUM?` might return -6.618E+6, indicating the minimum frequency drift is -6.618 MHz.

## **READ:FVTime[:MEASview<y>]:MINLocation? (Query Only)**

Returns the time at which the frequency drift is minimum.



## Conditions

Measurement views: Frequency versus Time

## Group

Read commands

## Syntax

```
READ:FVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
READ:FVTime[:MEASview<y>]:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<Nrf> is the time in seconds at which the frequency drift is minimum.

## Examples

READ:FVTIME:MEASview1:MINLOCATION? might return 450.7E-9, indicating the frequency drift is minimum at 450.7 ns.

## READ:FVTime[:MEASview<y>]:RESult? (Query Only)

Returns the Frequency versus Time measurement results.

## Conditions

Measurement views: Frequency versus Time

## Group

Read commands

## Syntax

```
READ:FVTime[:MEASview<y>]:RESult?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max>, <max\_time>, <min>, <min\_time>

Where

<max>::<Nrf> is the maximum frequency drift in Hz.

<max\_time>::<NRf> is the time in seconds at which the frequency drift is maximum.

<min>::<NRf> is the minimum frequency drift in Hz.

<min\_time>::<NRf> is the time in seconds at which the frequency drift is minimum.

## Examples

READ:FVTIME:MEASview1:RESULT? might return 2.625E+6,25.03E-9,-6.618E+6,450.7E-9, indicating the maximum frequency drift is 2.625 MHz at 25.03 ns and the minimum frequency drift is -6.618 MHz at 450.7 ns.

## READ:IQVTime[:MEASview<y>]:I? (Query Only)

Returns the I versus Time trace data.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Read commands

### Syntax

READ:IQVTime[:MEASview<y>]:I?

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the I level data for the point n in volts,

4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

READ:IQVTIME:MEASview1:I? might return #3160xxxx... (160-byte data) for the I versus Time trace.

## READ:IQVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value in the RF I & Q versus Time measurement.

### Conditions

Measurement views: RF I & Q versus Time

## Group

Read commands

## Syntax

```
READ:IQVTime[:MEASview<y>]:MAXimum?
```

## Related Commands

```
READ:IQVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max>: : <NRf> is the maximum I or Q level in volts.

## Examples

```
READ:IQVTIME:MEASview1:MAXIMUM? might return 1.214, indicating the maximum I or Q level is 1.214 V.
```

## READ:IQVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the I or Q level is maximum.

## Conditions

Measurement views: RF I & Q versus Time

## Group

Read commands

## Syntax

```
READ:IQVTime[:MEASview<y>]:MAXLocation?
```

## Related Commands

```
READ:IQVTime[:MEASview<y>]:MAXimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max\_time>: : <NRf> is the time in seconds at which the I or Q level is maximum.

## Examples

`READ:IQVTIME:MEASview1:MAXLOCATION?` might return `175.3E-9`, indicating the I or Q level is maximum at 175.3 ns.

## **READ:IQVTime[:MEASview<y>]:MINimum? (Query Only)**

Returns the minimum value in the RF I & Q versus Time measurement.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Read commands

### Syntax

`READ:IQVTime[:MEASview<y>]:MINimum?`

### Related Commands

`READ:IQVTime[:MEASview<y>]:MINLocation?`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

<min>::<NRf> is the minimum I or Q level in volts.

## Examples

`READ:IQVTIME:MEASview1:MINIMUM?` might return `-370.5E-3`, indicating the minimum I or Q level is -370.5 mV.

## **READ:IQVTime[:MEASview<y>]:MINLocation? (Query Only)**

Returns the time at which the I or Q level is minimum.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Read commands

### Syntax

`READ:IQVTime[:MEASview<y>]:MINLocation?`

## Related Commands

READ:IQVTime[:MEASview<y>]:MINimum?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<Nrf> is the time in seconds at which the I or Q level is minimum.

## Examples

READ:IQVTIME:MEASview1:MINLOCATION? might return 450.7E-9, indicating the I or Q level is minimum at 450.7 ns.

## READ:IQVTime[:MEASview<y>]:Q? (Query Only)

Returns the Q versus Time trace data.

## Conditions

Measurement views: IQ versus Time

## Group

Read commands

## Syntax

READ:IQVTime[:MEASview<y>]:Q?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the Q level data for the point n in volts,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:IQVTIME:MEASview1:Q? might return #3160xxxx... (160-byte data) for the Q versus Time trace.

## **READ:IQVTime[:MEASview<y>]:RESult? (Query Only)**

Returns the RF I & Q versus Time measurement results.

### **Conditions**

Measurement views: RF I & Q versus Time

### **Group**

Read commands

### **Syntax**

```
READ:IQVTime[:MEASview<y>]:RESult?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Returns**

```
<max>,<max_time>,<min>,<min_time>
```

Where

<max>::<NRf> is the maximum I or Q level in volts.

<max\_time>::<NRf> is the time in seconds at which the I or Q level is maximum.

<min>::<NRf> is the minimum I or Q level in volts.

<min\_time>::<NRf> is the time in seconds at which the I or Q level is minimum.

### **Examples**

```
READ:IQVTIME:MEASview1:RESULT? might return
```

```
1.214,175.3E-9,-370.5E-3,450.7E-9, indicating
```

the maximum I or Q level is 1.214 V at 175.3 ns and

the minimum I or Q level is -370.5 mV at 450.7 ns.

## **READ:LTE:ACLR:MHITs:COUNT? (Query Only)**

Returns the number of rows in the results table of the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

### **Group**

Read commands

## Syntax

```
READ:LTE:ACLR:MHITS:COUNT?
```

## Returns

```
<NR1>
```

## Examples

READ:LTE:ACLR:MHITS:COUNT? might return 7, indicating there are seven rows in the results table.

## READ:LTE:ACLR:MHITS<x>:BANDwidth? (Query Only)

Returns the bandwidth value of the channel (row) of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

```
READ:LTE:ACLR:MHITS<x>:BANDwidth?
```

## Returns

```
<NR3>
```

## Examples

READ:LTE:ACLR:MHITS1:BANDWIDTH? might return 4.515000000E+6, which indicates a bandwidth for channel 1 (row 1) of 4.515 MHz.

## READ:LTE:ACLR:MHITS<x>:CHANnel:NAME? (Query Only)

Returns the channel name of the specified index in the results table of the LTE ACLR display. The minimum index will be 1 and the maximum index will be the count of the channels in the results table.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

READ:LTE:ACLR:MHITs<x>:CHANnel:NAME?

## Returns

<string>

## Examples

READ:LTE:ACLR:MHITS2:CHANNEL:NAME? might return "B", which is the name of channel 2 (row 2).

## READ:LTE:ACLR:MHITs<x>:FREQuency:OFFSet? (Query Only)

Returns the frequency offset of the channel of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

READ:LTE:ACLR:MHITs<x>:FREQuency:OFFSet?

## Returns

<NR3>

## Examples

READ:LTE:ACLR:MHITS1:FREQUENCY:OFFSET? might return 5.000000000E+6, which indicates the frequency offset of channel 1 (row 1) is 5 MHz.

## READ:LTE:ACLR:MHITs<x>:INTEg:LOWEr:ABSolute? (Query Only)

Returns the lower absolute value of the channel of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

READ:LTE:ACLR:MHITs<x>:INTEg:LOWEr:ABSolute?



## Returns

<NRf>

## Examples

`READ:LTE:ACLR:MHITs1:INTEg:LOWEr:ABSolute?` might return -53.3921980303 which indicates lower absolute value of channel 1 (row 1) is -53.3921980303 dBm.



**Note:** The actual units depends on the choice made from the units tab of the Analysis control panel.

## **READ:LTE:ACLR:MHITs<x>:INTEg:LOWEr:RELative? (Query Only)**

Returns the lower relative value of the channel of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

`READ:LTE:ACLR:MHITs<x>:INTEg:LOWEr:RELative?`

## Returns

<NRf>

## Examples

`READ:LTE:ACLR:MHITs1:INTEg:LOWEr:RELATIVE?` might return -53.3921980303 which indicates lower relative value of channel 1 (row 1) is -53.3921980303 dB.

## **READ:LTE:ACLR:MHITs<x>:INTEg:UPPEr:ABSolute? (Query Only)**

Returns the upper absolute value of the channel of the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> is a positive integer and represents the row number in the LTE ACLR display results table.

## Group

Read commands

## Syntax

`READ:LTE:ACLR:MHITs<x>:INTEg:UPPEr:ABSolute?`

## Returns

<NRf>

## Examples

READ:LTE:ACLR:MHITS1:INTEG:UPPER:ABSOLUTE? might return -53.3921980303 which indicates upper absolute value of channel 1 (row 1) is -53.3921980303 dBm.



**Note:** The actual units depends on the choice made from the units tab of the Analysis control panel.

## READ:LTE:ACLR:REFerence:POWer? (Query Only)

Returns the reference power level in the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Read commands

## Syntax

READ:LTE:ACLR:REFerence:POWer?

## Returns

<NRf>

## Examples

READ:LTE:ACLR:REFERENCE:POWER? might return -16.92, indicating that the reference power level is -16.92 dBm.

## READ:LTE:ACLR:RESUlts:STATUs? (Query Only)

Returns the pass or fail status of the LTE ACLR display measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Read commands

## Syntax

READ:LTE:ACLR:RESUlts:STATUs?

## Returns

0 means measurement status is FAIL.

1 means measurement status is PASS.

## Examples

READ:LTE:ACLR:RESULTS:STATUS? might return 1, indicating that the LTE ACLR measurement status is PASS.

## READ:LTE:ACLR:SPECTrum:X? (Query Only)

Returns the frequency of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Read commands

### Syntax

READ:LTE:ACLR:SPECTrum:X?

### Returns

#<num\_digit><num\_byte><x(1)><x(2)>...<x(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the frequency (Hz) at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

READ:LTE:ACLR:SPECTRUM:X? might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace.

## READ:LTE:ACLR:SPECTrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Read commands

### Syntax

READ:LTE:ACLR:SPECTrum:XY?

### Returns

#<num\_digit><num\_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x (n) ><y (n) > is the frequency (Hz) and amplitude (dBm) pair at the nth point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:LTE:ACLR:SPECTRUM:XY?` might return `#516020xxxx...` (16020-byte data) for the frequency and amplitude pairs of the spectrum trace.

## **READ:LTE:ACLR:SPECTrum:Y? (Query Only)**

Returns the amplitude of the spectrum trace in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Read commands

### Syntax

`READ:LTE:ACLR:SPECTrum:Y?`

### Returns

`#<num_digit><num_byte><y (1) ><y (2) >...<y (n) >`

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y (n) > is the frequency (Hz) at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:LTE:ACLR:SPECTRUM:Y?` might return `#516020xxxx...` (16020-byte data) for the amplitude of the spectrum trace.

## **READ:LTE:CHSPectrum:OBW? (Query Only)**

Returns the Occupied Bandwidth (Hz) of the LTE Channel Spectrum measurement view.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Read commands

## Syntax

```
READ:LTE:CHSPepectrum:OBW?
```

## Returns

```
<NR3>
```

## Examples

READ:LTE:CHSPECTRUM:OBW? might return 1.0851851250E+6, indicating the occupied Bandwidth is 1.09 MHz.

## READ:LTE:CHSPepectrum:POWer:CHANnel? (Query Only)

Returns the Channel Power value in LTE Channel Spectrum display. The result will be in dBm (by default) or in the units chosen from the Units tab of the Analysis Control panel.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Read commands

## Syntax

```
READ:LTE:CHSPepectrum:POWer:CHANnel?
```

## Returns

```
<NRf>
```

## Examples

READ:LTE:CHSPECTRUM:POWER:CHANNEL? might return -14.9248560147, indicating that the Channel Power is -14.92 dBm.

## READ:LTE:CHSPepectrum:SPECTrum? (Query Only)

Returns spectrum trace data of the LTE Channel Spectrum measurement.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Read commands

## Syntax

```
READ:LTE:CHSPepectrum:SPECTrum?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

### Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of the data that follow.

<data (n)> is the amplitude in dBm at the nth data point, 4-byte little endian floating point format specified in IEEE 488.2.

### Examples

READ:LTE:CHSPECTRUM:SPECTRUM? might return #43204 xxxx... (3204-byte data) for the spectrum trace data of the LTE Channel Spectrum measurement.

## READ:LTE:CONStE:CELL:ID? (Query Only)

Queries the Cell ID value LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Read commands

### Syntax

READ:LTE:CONStE:CELL:ID?

### Returns

<NR1>

### Examples

READ:LTE:CONStE:CELL:ID? might return 112, indicating CELLID measurement number is 112.

## READ:LTE:CONStE:FREQuency:ERRor? (Query Only)

Queries Frequency Error value in Hz for the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Read commands

### Syntax

READ:LTE:CONStE:FREQuency:ERRor?

### Returns

<NRf>

## Examples

`READ:LTE:CONStE:FREQUENCY:ERROR?` might return `-71.9780578613`, indicating frequency error value is `-71.9780578613` Hz.

## **READ:LTE:CONStE:GROUp:ID? (Query Only)**

Queries the Group ID value for the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Read commands

### Syntax

`READ:LTE:CONStE:GROUp:ID?`

### Returns

<NR1>

## Examples

`READ:LTE:CONStE:GROUp:ID?` might return `160`, indicating Group ID measurement number is `160`.

## **READ:LTE:CONStE:SECTOr:ID? (Query Only)**

Queries the Sector ID value for the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Read commands

### Syntax

`READ:LTE:CONStE:SECTOr:ID?`

### Returns

<NRf>

## Examples

`READ:LTE:CONStE:SECTOr:ID?` might return `2`, indicating the Sector ID value is `2`.

## **READ:LTE:PVTime:OFFSlot:POWer? (Query Only)**

Returns the TOff power measurement value in dBm/MHz for the LTE Power vs Time display.

### **Conditions**

Measurement view: LTE Power vs Time

### **Group**

Read commands

### **Syntax**

READ:LTE:PVTime:OFFSlot:POWer?

### **Returns**

<NRf>

### **Examples**

READ:LTE:PVTIME:OFFSLOT:POWER? might return -76.11514587403, indicating the offslot power is -76.11514 dBm/MHz.

## **READ:LTE:PVTime:RESUlts:STATUs**

Returns the pass or fail status for the LTE Power vs Time measurement.

### **Conditions**

Measurement view: LTE Power vs Time

### **Group**

Read commands

### **Syntax**

READ:LTE:PVTime:RESUlts:STATUs

### **Returns**

0: means measurement status is FAIL.

1: means measurement status is PASS.

### **Examples**

READ:LTE:PVTIME:RESULTS:STATUS might return 1, indicating the measurement status is PASS.

## **READ:LTE:PVTime:TRACe:X? (Query Only)**

Returns the horizontal values (time in seconds) for the LTE Power vs. Time trace.



## Conditions

Measurement view: LTE Power vs Time

## Group

Read commands

## Syntax

```
READ:LTE:PVTime:TRACe:X?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:LTE:PVTIME:TRACE:X? might return #43204 (3204-byte data), which represents the time in seconds (horizontal values).

## READ:LTE:PVTime:TRACe:XY? (Query Only)

Returns the horizontal value (time in seconds) and vertical value (power) for the LTE Power vs. Time trace.

## Conditions

Measurement view: LTE Power vs Time

## Group

Read commands

## Syntax

```
READ:LTE:PVTime:TRACe:XY?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:LTE:PVTIME:TRACE:XY?` might return #43204 (3204-byte data), which represents the horizontal value (time in seconds) and vertical value (power) pair at the nth data point.

## **READ:LTE:PVTime:TRACe:Y? (Query Only)**

Returns the vertical values (power) for the LTE Power vs. Time trace.

### Conditions

Measurement view: LTE Power vs Time

### Group

Read commands

### Syntax

`READ:LTE:PVTime:TRACe:Y?`

### Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where,

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:LTE:PVTIME:TRACE:Y?` might return #43204 (3204-byte data), which represents the vertical values (power).

## **READ:MCPower:ADJacent:CHANnels? (Query Only)**

Returns the power of adjacent channels in order of increasing frequency.

### Conditions

Measurement views: MCPR

### Group

Read commands

### Syntax

`READ:MCPower:ADJacent:CHANnels?`

### Arguments

None

## Returns

<acpr\_lower (n)>, ...<acpr\_lower (2)>, <acpr\_lower (1)>, <acpr\_upper (1)>, <acpr\_upper (2)>, ...<acpr\_upper (n)>

Where

<acpr\_lower (n)> is the ACPR for the lower channel #n in dB.

<acpr\_upper (n)> is the ACPR for the upper channel #n in dB.

To add a pair of upper and lower adjacent channels, use the [SENSe]:MCPower:CHANnel:ADJacent:ADD command.

## Examples

READ:MCPOWER:ADJACENT:CHANNELS? might return

-4.420,-4.847,-4.316,-4.225, indicating

(ACPR for the lower channel 2) = -4.420 dB,

(ACPR for the lower channel 1) = -4.847 dB,

(ACPR for the upper channel 1) = -4.316 dB, and

(ACPR for the upper channel 2) = -4.225 dB.

## READ:MCPower:CHANnel:POWER? (Query Only)

Returns the reference power in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Read commands

## Syntax

READ:MCPower:CHANnel:POWER?

## Arguments

None

## Returns

<ref\_power>: <NRf> is the reference power in dBm. The unit can be changed by the [SENSe]:POWER:UNITs command.

To select the power reference, use the [SENSe]:MCPower:RChannels commands.

## Examples

READ:MCPOWER:CHANNEL:POWER? might return 4.227, indicating that the reference power is 4.227 dBm.

## **READ:MCPower:MAIN:CHANnels? (Query Only)**

Returns the power of main channels in order of increasing frequency.

### **Conditions**

Measurement views: MCPR

### **Group**

Read commands

### **Syntax**

READ:MCPower:MAIN:CHANnels?

### **Related Commands**

[ :SENSe ] :MCPower:CHANnel:MAIN commands

### **Arguments**

None

### **Returns**

<power\_main(1)>, <power\_main(2)>, ... <power\_main(n)>

Where

<power\_main(n)> is the power of main channel #n in dBm. The unit can be changed by the [SENSe]:POWER:UNITs command.

To specify the main channels, use the [SENSe]:MCPower:CHANnel:MAIN commands.

### **Examples**

READ:MCPOWER:MAIN:CHANNELS? might return

-2.420,-2.847,-2.316,-2.225, indicating

(power of the main channel 1) = -2.420 dBm,

(power of the main channel 2) = -2.847 dBm,

(power of the main channel 3) = -2.316 dBm, and

(power of the main channel 4) = -2.225 dBm.

## **READ:MCPower:SPECTrum? (Query Only)**

Returns spectrum trace data of the MCPR measurement.

### **Conditions**

Measurement views: MCPR

## Group

Read commands

## Syntax

```
READ:MCPower:SPECTrum?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the spectrum trace data in dBm for the point n,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:MCPOWER:SPECTRUM? might return #43204xxxx... (3204-byte data) for the spectrum trace data of the MCPR measurement.

## READ:MERRor[:MEASview<y>]:FERRor? (Query only)

Returns the frequency error in the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Read commands

## Syntax

```
READ:MERRor[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`READ:MERRor:MEASview2:FERRor?` might return `-10.7E+3`, indicating the frequency error is `-10.7 kHz`.

## **READ:MERRor[:MEASview<y>]:PEAK? (Query only)**

Returns the peak value in the Magnitude error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Read commands

## Syntax

`READ:MERRor[:MEASview<y>]:PEAK?`

## Related Commands

[\*READ:MERRor\[:MEASview<y>\]:PINdex?\*](#)

## Arguments

None

## Returns

`<peak>::<NRf>` is the peak magnitude error value in percent (%).

## Examples

`READ:MERRor:MEASview2:PEAK?` might return `1.57`, indicating the peak magnitude error is `1.57%`.

## **READ:MERRor[:MEASview<y>]:PINdex? (Query only)**

Returns the time at the Magnitude error peak.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Read commands

## Syntax

```
READ:MERRor[:MEASview<y>]:PINdex?
```

## Related Commands

[READ:MERRor\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

<peak\_time>::<NRf> is the time at the Magnitude error peak in symbol number. The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

READ:MERRor:MEASview2:PINdex? might return 68.000, indicating that the Magnitude error peak is at symbol #68.

## READ:MERRor[:MEASview<y>]:RMS? (Query only)

Returns the RMS (Root-Mean-Square) value in the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time.

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Read commands

## Syntax

```
READ:MERRor[:MEASview<y>]:RMS?
```

## Arguments

None

## Returns

<rms>::<NRf> is the RMS magnitude error value in percent (%).

## Examples

`READ:MERRor:MEASview2:RMS?` might return 0.382, indicating the RMS MERRor value is 0.382% .

## **READ:MERRor[:MEASview<y>]:TRACe? (Query only)**

Returns the Magnitude error versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Read commands

## Syntax

`READ:MERRor[:MEASview<y>]:TRACe?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the Magnitude error versus Time trace data for the point n in percent (%),

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:MERRor:MEASview2:TRACe?` might return `#42036xxxx...` (2036-byte data) for the Magnitude error versus Time measview 2.

## **READ:NOISe:TABLE:VALue? (Query Only)**

Returns the noise table data. A total of 7 values per frequency (28 bytes per frequency) is returned. The number of frequencies returned depends on the settings.

## Conditions

Measurement view: Noise display



## Group

Read commands

## Syntax

```
READ:NOISE:TABLE:VALUE?
```

## Returns

Returns an array of floats (size of 4 bytes each) defined as:

- First frequency gain, noise figure, noise factor, y factor, noise temperature, power hot, power cold
- Second frequency gain, noise figure, noise factor, y factor, noise temperature, power hot, power cold

## Examples

```
READ:NOISE:TABLE:VALUE?
```

## READ:OBWidth:BOBW:XDBBandwidth? (Query Only)

Returns the difference between the higher and lower frequency points corresponding to the value where it is X dB less from the value at the center frequency.

Set the measurement direction using the appropriate inward or outward command before issuing this command.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Read commands

## Syntax

```
READ:OBWidth:BOBW:XDBBandwidth?
```

## Related Commands

```
READ:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?,  
READ:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth?
```

## Arguments

None

## Returns

<value> ::= <NRf> the X dBm bandwidth in Hz.

## Examples

READ:OBWIDTH:BOBW:XDBBANDWIDTH? might return 89.12000E+3 indicating the difference between the higher and lower frequency points is 89.12 kHz.

## **READ:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth? (Query Only)**

Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured inwards (coming from the ends to the center frequency).

### **Conditions**

Measurement views: Bluetooth 20 dB Bandwith

### **Group**

Read commands

### **Syntax**

READ:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?

### **Related Commands**

READ:OBWidth:BOBW:XDBBANDwidth?,  
READ:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQUENCY?

### **Arguments**

None

### **Returns**

<value> ::= <NRF> the input X dBm value in Hz.

### **Examples**

READ:OBWIDTH:BOBW:XDBMBANDWIDTH:IN:BANDWIDTH? might return 1.251840000E+6 indicating that the bandwidth corresponding to the input XdBm vaue is 1.2518 MHz measured in the inward direction.

## **READ:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQUENCY? (Query Only)**

Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the inward direction.

### **Conditions**

Measurement views: Bluetooth 20 dB Bandwith

### **Group**

Read commands

### **Syntax**

READ:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQUENCY?

### **Related Commands**

READ:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQUENCY?

## Arguments

None

## Returns

<value> ::= <NRf> the left frequency in Hz.

## Examples

READ:OBWIDTH:BOBW:XDBMBANDWIDTH:IN:LEFT:FREQUENCY? might return 2.401E+9 indicating that the frequency corresponding to the X dbm value to the left of the center frequency is 2.401GHz.

## READ:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQUENCY? (Query Only)

Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the inward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:BOBW:XDBMbandwidth:IN:RIGHT:FREQUENCY?

## Related Commands

READ:OBWidth:BOBW:XDBMbandwidth:IN:LEFT:FREQUENCY?

## Arguments

None

## Returns

<value> ::= <NRf> the right frequency in Hz.

## Examples

READ:OBWIDTH:BOBW:XDBMBANDWIDTH:IN:RIGHT:FREQUENCY? might return 2.445E+9 indicating that the frequency corresponding to the XdBm value to the right of the center the frequency is 2.445 GHz.

## READ:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth? (Query Only)

Returns the difference between the higher and lower frequency points corresponding to the value which it is the X dBm value measured outwards (from the center frequency to the ends).

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:BOBW:XDBMbandwidth:OUT:BANDwidth?

## Related Commands

READ:OBWidth:BOBW:XDBBANDwidth?, READ:OBWidth:BOBW:XDBMbandwidth:IN:BANDwidth?

## Arguments

None

## Returns

<value> ::= <NRf> the input X dBm value in Hz.

## Examples

READ:OBWIDTH:BOBW:XDBMBANDWIDTH:OUT:BANDWIDTH? might return 1.251840000E+6 indicating that the bandwidth corresponding to the output XdBm value is 1.2518 MHz measured in the outward direction.

## READ:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency? (Query Only)

Returns the frequency corresponding to the X dBm value to the left of the center frequency measured in the outward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwith

## Group

Read commands

## Syntax

READ:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency?

## Related Commands

READ:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency?

## Arguments

None

## Returns

<value> ::= <NRf> the left frequency in Hz.

## Examples

`READ:OBWIDTH:BOBW:XDBMBANDWIDTH:OUT:LEFT:FREQUENCY?` might return 2.401E+9 indicating that the frequency corresponding to the X dbm value to the left of the center frequency is 2.401GHz.

## **READ:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency? (Query Only)**

Returns the frequency corresponding to the X dBm value to the right of the center frequency measured in the outward direction.

## Conditions

Measurement views: Bluetooth 20 dB Bandwidth

## Group

Read commands

## Syntax

`READ:OBWidth:BOBW:XDBMbandwidth:OUT:RIGHT:FREQuency?`

## Related Commands

`READ:OBWidth:BOBW:XDBMbandwidth:OUT:LEFT:FREQuency?`

## Arguments

None

## Returns

`<value>` ::= `<NRf>` the Right frequency in Hz.

## Examples

`READ:OBWIDTH:BOBW:XDBMBANDWIDTH:OUT:RIGHT:FREQUENCY?` might return 2.445E+9 indicating that the frequency corresponding to the XdBm value to the right of the center the frequency is 2.445 GHz.

## **READ:OBWidth:FREQuency:ERRor? (Query Only)**

Returns the frequency error in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

`READ:OBWidth:FREQuency:ERRor?`

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`READ:OBWIDTH:FREQUENCY:ERROR?` might return `-10.7E+3`, indicating the frequency error is -10.7 kHz.

## **READ:OBWidth:OBWidth:BANDwidth? (Query Only)**

Returns the occupied bandwidth in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

`READ:OBWidth:OBWidth:BANDwidth?`

## Arguments

None

## Returns

`<OBW>::<NRf>` is the occupied bandwidth in Hz.

## Examples

`READ:OBWIDTH:OBWIDTH:BANDWIDTH?` might return `4.0E+6`, indicating the occupied bandwidth is 4 MHz.

## **READ:OBWidth:OBWidth:LEFT:FREQuency? (Query Only)**

Returns the left (lower) frequency of the occupied bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

`READ:OBWidth:OBWidth:LEFT:FREQuency?`

---

## Related Commands

READ:OBWidth:OBWidth:RIGHT:FREQUENCY?

## Arguments

None

## Returns

<OBW\_left\_freq>::<NRf> is the left frequency in Hz.

## Examples

READ:OBWIDTH:OBWIDTH:LEFT:FREQUENCY?

## READ:OBWidth:OBWidth:LEFT:LEVel? (Query Only)

Returns the level at the left frequency of the occupied bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:OBWidth:LEFT:LEVel?

## Related Commands

READ:OBWidth:OBWidth:RIGHT:LEVel?

## Arguments

None

## Returns

<OBW\_left\_level>::<NRf> is the level at the left frequency in dB.

## Examples

READ:OBWIDTH:OBWIDTH:LEFT:LEVEL? might return -23.5, indicating the level at the left frequency is -23.5 dB.

## READ:OBWidth:OBWidth:POWer? (Query Only)

Returns the reference power in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:OBWidth:POWer?

## Arguments

None

## Returns

<OBW\_ref\_power>::<NRf> is the reference power in dBm.

The unit can be changed by the [SENSE]:POWer:UNITs command.

## Examples

READ:OBWIDTH:OBWIDTH:POWER? might return -10.0, indicating the reference power is -10 dBm.

## READ:OBWidth:OBWidth:RIGHT:FREQUENCY? (Query Only)

Returns the right (higher) frequency of the occupied bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:OBWidth:RIGHT:FREQUENCY?

## Related Commands

READ:OBWidth:OBWidth:LEFT:FREQUENCY?

## Arguments

None

## Returns

<OBW\_right\_freq>::<NRf> is the right frequency in Hz.

## Examples

READ:OBWIDTH:OBWIDTH:RIGHT:FREQUENCY? might return 1.502E+9, indicating the right frequency is 1.502 GHz.



## **READ:OBWidth:OBWidth:RIGHT:LEVel? (Query Only)**

Returns the level at the right frequency of the occupied bandwidth.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Read commands

### **Syntax**

```
READ:OBWidth:OBWidth:RIGHT:LEVel?
```

### **Related Commands**

```
READ:OBWidth:OBWidth:LEFT:LEVel?
```

### **Arguments**

None

### **Returns**

<OBW\_right\_level>::<NRf> is the level at the right frequency in dB.

### **Examples**

READ:OBWIDTH:OBWIDTH:RIGHT:LEVEL? might return -23.5, indicating the level at the right frequency is -23.5 dB.

## **READ:OBWidth:SPECTrum? (Query Only)**

Returns spectrum trace data of the Occupied Bandwidth measurement.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Read commands

### **Syntax**

```
READ:OBWidth:SPECTrum?
```

### **Arguments**

None

### **Returns**

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n) > is the amplitude in dBm at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:OBWIDTH:SPECTRUM? might return #43204xxxx... (3204-byte data) for the spectrum trace data of the Occupied Bandwidth measurement.

## READ:OBWidth:XDBBandwidth:BANDwidth? (Query Only)

Returns the x dB bandwidth in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Read commands

### Syntax

READ:OBWidth:XDBBandwidth:BANDwidth?

### Arguments

None

### Returns

<xdB BW> : : <NRF> is the x dB bandwidth in Hz.

## Examples

READ:OBWIDTH:XDBBANDWIDTH:BANDWIDTH? might return 2.0E+6, indicating the x dB bandwidth is 2 MHz.

## READ:OBWidth:XDBBandwidth:LEFT:FREQuency? (Query Only)

Returns the left (lower) frequency of the x dB bandwidth.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Read commands

## Syntax

```
READ:OBWidth:XDBBandwidth:LEFT:FREQUENCY?
```

## Related Commands

```
READ:OBWidth:XDBBandwidth:RIGHT:FREQUENCY?
```

## Arguments

None

## Returns

<xdBW\_left\_freq>::<Nrf> is the left frequency in Hz.

## Examples

READ:OBWIDTH:XDBBANDWIDTH:LEFT:FREQUENCY? might return 1.498E+9, indicating the left frequency is 1.498 GHz.

## READ:OBWidth:XDBBandwidth:LEFT:LEVEL? (Query Only)

Returns the level at the left frequency of the x dB bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

```
READ:OBWidth:XDBBandwidth:LEFT:LEVEL?
```

## Related Commands

```
READ:OBWidth:XDBBandwidth:RIGHT:LEVEL?
```

## Arguments

None

## Returns

<xdBW\_left\_level>::<Nrf> is the level at the left frequency in dB.

## Examples

READ:OBWIDTH:XDBBANDWIDTH:LEFT:LEVEL? might return -23.5, indicating the level at the left frequency is -23.5 dB.

## READ:OBWidth:XDBBandwidth:POWER? (Query Only)

Returns the reference power in the x dB bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:XDBBandwidth:POWer?

## Arguments

None

## Returns

<xdBW\_ref\_power>::<Nrf> is the reference power in dBm. The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:OBWIDTH:XDBBANDWIDTH:POWER? might return -10.0, indicating the reference power is -10 dBm.

## READ:OBWidth:XDBBandwidth:RIGHT:FREQuency? (Query Only)

Returns the right (higher) frequency of the x dB bandwidth.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Read commands

## Syntax

READ:OBWidth:XDBBandwidth:RIGHT:FREQuency?

## Related Commands

READ:OBWidth:XDBBandwidth:LEFT:FREQuency?

## Arguments

None

## Returns

<xdBW\_right\_freq>::<Nrf> is the right frequency in Hz.

## Examples

READ:OBWIDTH:XDBBANDWIDTH:RIGHT:FREQUENCY? might return 1.502E+9, indicating the right frequency is 1.502 GHz.

## **READ:OBWidth:XDBBandwidth:RIGHT:LEVEL? (Query Only)**

Returns the level at the right frequency of the x dB bandwidth.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Read commands

### **Syntax**

```
READ:OBWidth:XDBBandwidth:RIGHT:LEVEL?
```

### **Related Commands**

```
READ:OBWidth:XDBBandwidth:LEFT:LEVEL?
```

### **Arguments**

None

### **Returns**

<xdbbw\_right\_level>::<Nrf> is the level at the right frequency in dB.

### **Examples**

READ:OBWIDTH:XDBBANDWIDTH:RIGHT:LEVEL? might return -23.5, indicating the level at the right frequency is -23.5 dB.

## **READ:OFDM:APOWER? (Query Only)**

Returns the average power in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:APOWER?
```

### **Arguments**

None

### **Returns**

<apower>::<Nrf> the average power in dB.

## Examples

READ:OFDM:APOWER? might return -23.4584459235 indicating the average power is -23.46 dB.

## READ:OFDM:APOWER:PEAK? (Query Only)

Returns the peak-to-average power in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:APOWER:PEAK?

## Arguments

None

## Returns

<apower\_peak>::<Nrf> is the peak-to-average power in dBm.

## Examples

READ:OFDM:APOWER:PEAK? might return 10.4140096289 indicating that peak-to-average power is 10.41 dBm.

## READ:OFDM:CONStE:MAGNitude? (Query Only)

Returns the constellation magnitude data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:CONStE:MAGNitude?

## Related Commands

READ:OFDM:CONStE:PHASe?

READ:OFDM:CONStE:TYPE?

READ:OFDM:CONStE:VALue?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:CONStE:MAGNITUDE? might return #510400xxx... (10400-byte data) for the measurement.

## READ:OFDM:CONStE:PHASe? (Query Only)

Returns the constellation phase data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:CONStE:PHASe?

## Related Commands

READ:OFDM:CONStE:MAGNitude?

READ:OFDM:CONStE:TYPE?

READ:OFDM:CONStE:VALue?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:CONStE:PHASe? might return #510400xxxx... (10400-byte data) for the measurement.

## READ:OFDM:CONStE:TYPE? (Query Only)

Returns the constellation context value of the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:CONStE:TYPE?

## Related Commands

READ:OFDM:CONStE:MAGNitude?

READ:OFDM:CONStE:PHASe?

READ:OFDM:CONStE:VALue?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers. The integers must be translated to the proper context as listed below.

Integer value	Type (context)
0	Pilot
1	Data
2	Unused or Null

## Examples

READ:OFDM:CONStE:TYPE? might return #41352 (1352-byte data) for the measurement. The actual data must be decoded to the context type using the table above.



## READ:OFDM:CONStE:VALue? (Query Only)

Returns the constellation value of the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Read commands

### Syntax

```
READ:OFDM:CONStE:VALue?
```

### Related Commands

```
READ:OFDM:CONStE:MAGNitude?
```

```
READ:OFDM:CONStE:PHASe?
```

```
READ:OFDM:CONStE:TYPE?
```

### Arguments

None

### Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers.

### Examples

READ:OFDM:CONStE:VALue? might return #41352 indicating 1352 bytes of data for the measurement.

## READ:OFDM:CPE? (Query Only)

Returns the Common Pilot Error magnitude for the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Read commands

### Syntax

```
READ:OFDM:CPE?
```

## Arguments

None

## Returns

<NRf> the RMS magnitude error in percent.

## Examples

READ:OFDM:CPE? might return 3.7868041505 indicating that the CPE is 3.787%.

## READ:OFDM:CRESPonse:MAGNitude? (Query Only)

Returns the channel response magnitude data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:CRESPonse:MAGNitude?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:CRESPONSE:MAGNITUDE? might return #3804xxxx... (804 byte data) for the measurement.

## READ:OFDM:CRESPonse:PHASe? (Query Only)

Returns the channel response phase data for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:CRESPonse:PHASe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:CRESPONSE:PHASE? might return #3804xxxx... (804 byte data) for the measurement.

## READ:OFDM:EVM:PEAK:DECibel:ALL? (Query Only)

Returns the peak EVM data for all subcarriers in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:EVM:PEAK:DECibel:ALL?
```

## Arguments

None

## Returns

<NRf> data in dB.

## Examples

READ:OFDM:EVM:PEAK:DECIBEL:ALL? might return -20.1872549032 indicating the data is -20.19 dB for the measurement.

## **READ:OFDM:EVM:PEAK:DECibel:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:PEAK:DECibel:DATA?`

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

`READ:OFDM:EVM:PEAK:DECIBEL:DATA?` might return `-20.1872549032` indicating the data is `-20.19` dB for the measurement.

## **READ:OFDM:EVM:PEAK:DECibel:PILOts? (Query Only)**

Returns the peak EVM data for the pilot subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:PEAK:DECibel:PILOts?`

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

`READ:OFDM:EVM:PEAK:DECIBEL:PILOTS?` might return `-20.2872549032` indicating the data is `-20.29` dB for the measurement.

## **READ:OFDM:EVM:PEAK:PERCent:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:PEAK:PERCent:ALL?
```

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

READ:OFDM:EVM:PEAK:PERCENT:ALL? might return 19.3223863840 indicating the data is 19.322% for the measurement.

## **READ:OFDM:EVM:PEAK:PERCent:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:PEAK:PERCent:DATA?
```

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

READ:OFDM:EVM:PEAK:PERCENT:DATA? might return 19.3223863840 indicating the data is 19.322% for the measurement.

## **READ:OFDM:EVM:PEAK:PERCent:PILOts? (Query Only)**

Returns the peak EVM data for the pilot subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:PEAK:PERCent:PILOts?`

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

`READ:OFDM:EVM:PEAK:PERCENT:PILOTS?` might return 10.8543863840 indicating the data is 10.854% for the measurement.

## **READ:OFDM:EVM:PEAK:SCARrier:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:PEAK:SCARrier:ALL?`

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

`READ:OFDM:EVM:PEAK:SCARRIER:ALL?` might return 99.00000 indicating the data is 99 for the measurement.

---

## **READ:OFDM:EVM:PEAK:SCARrier:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:PEAK:SCARrier:DATA?
```

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

READ:OFDM:EVM:PEAK:SCARRIER:DATA? might return 99.00000 indicating the data is 99 for the measurement.

## **READ:OFDM:EVM:PEAK:SCARrier:PILOts? (Query Only)**

Returns the peak EVM data for the pilot subcarriers at the subcarrier level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:PEAK:SCARrier:PILOts?
```

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

READ:OFDM:EVM:PEAK:SCARRIER:PILOTS? might return 88.00000 indicating the data is 88 for the measurement

## **READ:OFDM:EVM:PEAK:SYMBOL:ALL? (Query Only)**

Returns the peak EVM data for all subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

READ:OFDM:EVM:PEAK:SYMBOL:ALL?

### **Arguments**

None

### **Returns**

<NR1>

### **Examples**

READ:OFDM:EVM:PEAK:SYMBOL:ALL? might return 2.00000 indicating the data is 2 for the measurement

## **READ:OFDM:EVM:PEAK:SYMBOL:DATA? (Query Only)**

Returns the peak EVM data for the data subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

READ:OFDM:EVM:PEAK:SYMBOL:DATA?

### **Arguments**

<NR1>

### **Returns**

### **Examples**

READ:OFDM:EVM:PEAK:SYMBOL:DATA? might return 2.00000 indicating the data is 2 for the measurement



---

## **READ:OFDM:EVM:PEAK:SYMBOL:PILOTS? (Query Only)**

Returns the peak EVM data for the pilot subcarriers at the symbol level in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:PEAK:SYMBOL:PILOTS?
```

### **Arguments**

<NR1>

### **Returns**

### **Examples**

READ:OFDM:EVM:PEAK:SYMBOL:PILOTS? might return 1.00000 indicating the data is 1 for the measurement

## **READ:OFDM:EVM:RMS:DECIBEL:ALL? (Query Only)**

Returns the RMS EVM data for all subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:RMS:DECIBEL:ALL?
```

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

READ:OFDM:EVM:RMS:DECIBEL:ALL? might return -26.9012093267 indicating the data is -26.90 for the measurement.

## **READ:OFDM:EVM:RMS:DECibel:DATA? (Query Only)**

Returns the RMS EVM data for the data subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:RMS:DECibel:DATA?`

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

`READ:OFDM:EVM:RMS:DECIBEL:DATA?` might return `-26.8477116269` indicating the data is `-26.85` for the measurement.

## **READ:OFDM:EVM:RMS:DECibel:PILots? (Query Only)**

Returns the RMS EVM data for the pilot subcarriers in the OFDM measurement.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

`READ:OFDM:EVM:RMS:DECibel:PILots?`

### **Arguments**

None

### **Returns**

<NRf> data in dB.

### **Examples**

`READ:OFDM:EVM:RMS:DECIBEL:PILOTS?` might return `-33.0589143032` indicating the data is `-33.06` dB.

---

## **READ:OFDM:EVM:RMS:PERCent:ALL? (Query Only)**

Returns the peak RMS data for all subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:RMS:PERCent:ALL?
```

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

READ:OFDM:EVM:RMS:PERCENT:ALL? might return 225.0743627548 indicating the data is 225.074% for the measurement.

## **READ:OFDM:EVM:RMS:PERCent:DATA? (Query Only)**

Returns the peak RMS data for the data subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

```
READ:OFDM:EVM:RMS:PERCent:DATA?
```

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

READ:OFDM:EVM:RMS:PERCENT:DATA? might return 260.6973409653 indicating that the data is 260.697% for the measurement.

## **READ:OFDM:EVM:RMS:PERCent:PILots? (Query Only)**

Returns the peak RMS data for the pilot subcarriers in the OFDM measurement as a percent.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

READ:OFDM:EVM:RMS:PERCent:PILots?

### **Arguments**

None

### **Returns**

<NRf> data in percent.

### **Examples**

READ:OFDM:EVM:RMS:PERCENT:PILOTS? might return 210.8103863840 indicating the data is 210.810% for the measurement.

## **READ:OFDM:EVM:TRACe<x>? (Query Only)**

Returns the EVM trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

### **Conditions**

Measurement views: OFDM

### **Group**

Read commands

### **Syntax**

READ:OFDM:EVM:TRACe<x>?

### **Arguments**

None

### **Returns**

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:EVM:TRACE2? might return #252xxxx. . . (52 byte data) for the Average versus Symbols measurement.

## READ:OFDM:FERRor? (Query Only)

Returns the Frequency error reading for the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Read commands

### Syntax

READ:OFDM:FERRor?

### Arguments

None

### Returns

<NRf> frequency error.

## Examples

READ:OFDM:FERROR? might return 82.8617142098E+3 indicating the frequency error was 82.86 kHz.

## READ:OFDM:FLATness:PASS? (Query Only)

Queries whether the average power level values of subcarriers across the signal bandwidth on the OFDM Spectral Flatness display remain within the limits defined for a particular standard.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Read commands

## Syntax

`READ:OFDM:FLATness:PASS?`

## Returns

Returns "1" in the case that all data remains within the defined limits, and "0" otherwise.

## Examples

`READ:OFDM:FLAT:PASS?` might return 1, which means that the average power level values of subcarriers across the signal bandwidth remain within the defined limits.

## READ:OFDM:FLATness:RESult? (Query Only)

Returns the summary results of each segment of the OFDM Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Read commands

## Syntax

`READ:OFDM:FLATness:RESult?`

## Returns

The data is returned as 12 comma delineated text values. The specific values are :

- minimum subcarrier within segment 1
- minimum subcarrier average energy deviation from the global average within segment 1
- deviation the of minimum subcarrier average from the allowed value in segment 1
- maximum subcarrier within segment 1
- maximum subcarrier average energy deviation from the global average within segment 1
- deviation of the maximum subcarrier average from the allowed value in segment 1
- minimum subcarrier within segment 2
- minimum subcarrier average energy deviation from the global average within segment 2
- deviation the of minimum subcarrier average from the allowed value in segment 2
- maximum subcarrier within segment 2
- maximum subcarrier average energy deviation from the global average within segment 2
- deviation of the maximum subcarrier average from the allowed value in segment 2

## Examples

`READ:OFDM:FLAT:RES?` might return might return -6.0000000000,-17.0626174659E-3,3.9829373825,12.0000000000,19.9792608 which represents the summary results of each segment of the OFDM Spectral Flatness display.

## READ:OFDM:FLATness:TRACe<x>? (Query Only)

Returns the OFDM Spectral Flatness trace data. When <x> is 1, the parameter is Matrix (symbol deviation per subcarrier). When <x> is 2, the parameter is Average Deviation vs Subcarrier.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Read commands

### Syntax

```
READ:OFDM:FLATness:TRACe<x>?
```

### Returns

```
#<num_digit><num_byte><data (1)><data (2)>..<data (n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

### Examples

READ:OFDM:FLAT:TRACE2? might return #510400.. (10400-byte data) which represents the data for the Average Deviation vs. Subcarrier trace.

## READ:OFDM:GAIN:IMBalance? (Query Only)

Returns the gain imbalance for the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Read commands

### Syntax

```
READ:OFDM:GAIN:IMBalance?
```

### Arguments

None

## Returns

<NRf> gain imbalance in dB.

## Examples

READ:OFDM:GAIN:IMBALANCE? might return  $-57.746E-3$  indicating that the gain imbalance is  $-0.057746$  dB.

## READ:OFDM:IQ:ORIGIN:OFFSet? (Query Only)

Returns the IQ origin offset for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:IQ:ORIGIN:OFFSet?

## Arguments

None

## Returns

<NRf> gain imbalance in dB.

## Examples

READ:OFDM:IQ:ORIGIN:OFFSET? might return  $-53.47017$  indicating that the IQ origin offset is  $-53.47$  dB.

## READ:OFDM:MERRor:TRACe<x>? (Query Only)

Returns the magnitude error trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:MERRor:TRACe<x>?



## Arguments

None

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:MERROR:TRACE2 might return #3108xxxx... (108-byte data) for the average versus symbols trace measurement.

## READ:OFDM:PACKet:DIRection? (Query Only)

Returns the direction of the packet in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:PACKet:DIRection?

## Arguments

None

## Returns

One of the following values.

UP specifies that the packet is in the up direction.

DOWN specifies that the packet is in the down direction.

BOTH specifies that the packet is both directions.

## Examples

READ:OFDM:PACKET:DIRECTION? might return DOWN indicating that the packet was in the down direction.

## READ:OFDM:PERRor:TRACe<x>? (Query Only)

Returns the phase error trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:PERRor:TRACe<x>?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:PERRor:TRACE2 might return #3100xxxx. . . (100-byte data) for the average versus symbols trace measurement.

## READ:OFDM:POWer:TRACe<x>? (Query Only)

Returns the power trace data for the OFDM measurement.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbols.

When <x> is 3, the parameter is Average versus Subcarrier.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:POWer:TRACe<x>?
```

## Arguments

None

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:OFDM:POWER:TRACE2 might return #296xxxx... (96-byte data) for the average versus symbols trace measurement.

## READ:OFDM:QUADrature:OFFSet? (Query Only)

Returns the quadrature offset in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:QUADrature:OFFSet?

## Arguments

None

## Returns

<NRf> quadrature offset in degrees.

## Examples

READ:OFDM:QUADRATURE:OFFSET? might return -99.9 indicating that the offset is -99°.

## READ:OFDM:SCARriers? (Query Only)

Returns the number of subcarriers in the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:SCARriers?

## Arguments

None

## Returns

<NR1> number of subcarriers.

## Examples

READ:OFDM:SCARRIERS? might return 200.0000 indicating that there are 200 subcarriers in the measurement.

## READ:OFDM:SCARriers:SPACing? (Query Only)

Returns the subcarrier spacing for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:SCARriers:SPACing?

## Arguments

None

## Returns

<NRf> subcarrier spacing.

## Examples

READ:OFDM:SCARRIERS:SPACING? might return 90.000E+3 indicating that the subcarrier spacing is 90.000 kHz.

## READ:OFDM:STABLE:VALUe? (Query Only)

Returns the number of bytes in the symbol table for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:STABLE:VALUE?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte>
```

Where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data.

## Examples

READ:OFDM:STABLE:VALUE? might return #43848 indicating that there are 3848 bytes in the symbol table.

## READ:OFDM:SYMBOL:CERRor? (Query Only)

Returns the symbol clock error for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

```
READ:OFDM:SYMBOL:CERRor?
```

## Arguments

None

## Returns

<NRf> symbol clock error in ppm.

## Examples

READ:OFDM:SYMBOL:CERROR? might return 422.7135479929 indicating that the symbol clock error was 422.714 ppm.

## READ:OFDM:SYMBOL:COUNT? (Query Only)

Returns the number of symbols for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Read commands

## Syntax

READ:OFDM:SYMBOL:COUNT?

## Arguments

None

## Returns

<NR1> number of symbols in the measurement.

## Examples

READ:OFDM:SYMBOL:COUNT? might return 25.00000 indicating that there were 25 symbols for the measurement.

## READ:P25:CONStE:FERRor? (Query Only)

Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency of the analyzer.

When Frequency Error is in Auto mode, this query returns the frequency error value. When in Manual mode, it returns the frequency offset.

## Conditions

Measurement view: P25 Constellation

## Group

Read commands

## Syntax

READ:P25:CONStE:FERRor?

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

READ:P25:CONStE:FERRor? might return -10.7E+3, which is a frequency error of -10.7 kHz.

## READ:P25:EDIagram:FDEVIation? (Query Only)

Returns the Frequency Deviation vs Time trace data with the X values.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Read commands

## Syntax

```
READ:P25:EDIagram:FDEVIation?
```

## Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<Y(n)> and <X(n)> are the frequency deviation in Hz and time (symbols) coordinate pair at the nth data point. 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

READ:P25:EDIAGRAM:FDEVIATION? might return #3160xxx... (160-byte data) For the Frequency Deviation vs Time trace.

## READ:P25:EDIagram:FERRor? (Query Only)

Returns the frequency error in Hz. The frequency error is the difference between the measured carrier frequency of the signal and the user-selected center frequency of the analyzer.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Read commands

## Syntax

```
READ:P25:EDIagram:FERRor?
```

## Returns

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

READ:P25:EDIAGRAM:FERROR? might return -10.7E+3, which is a frequency error of -10.7 kHz.

## READ:P25:PVTime:BURSt:POWer? (Query Only)

Returns the average power of the burst packet, in dBm, for the P25 Power vs. Time trace.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Read commands

## Syntax

`READ:P25:PVTime:BURSt:POWer?`

## Returns

Floating point number that represents the average power of the burst packet, in dBm.

## Examples

`READ:P25:PVTIME:BURST:POWER?` might return -1.99, which represents the value of the average power of the burst packet of -1.99 dBm.

## **READ:P25:PVTime:FALL:TIME? (Query Only)**

Queries the value of the width of the falling edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Read commands

## Syntax

`READ:P25:PVTime:FALL:TIME?`

## Related Commands

`READ:P25:PVTime:FTIME?`

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

`READ:P25:PVTIME:FALL:TIME?` might return 7e-6, which indicates the width of the falling edge is 7  $\mu$ s.

## **READ:P25:PVTime:FTIME? (Query Only)**

Queries the value of the width of the falling edge, in seconds.

## Conditions

Measurement view: P25 Power vs. Time



## Group

Read commands

## Syntax

```
READ:P25:PVTime:FTIME?
```

## Related Commands

```
READ:P25:PVTime:FALL:TIME?
```

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

READ:P25:PVTIME:FTIME? might return 7e-6, which indicates the width of the falling edge is 7  $\mu$ s.

## READ:P25:PVTime:TRACe:X? (Query Only)

Returns the horizontal values (time in seconds) for the P25 Power vs. Time trace.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Read commands

## Syntax

```
READ:P25:PVTime:TRACe:X?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:P25:PVTIME:TRACE:X? might return #43204 (3204-byte data), which represents the value of the time in seconds (horizontal values) for the P25 Power vs. Time trace.

## READ:P25:PVTime:TRACe:XY? (Query Only)

Returns the horizontal value (time in seconds) and vertical value (power) for the P25 Power vs. Time trace <x(n)><y(n)> is the horizontal value and vertical value pair at the nth data point.

## Conditions

Measurement view: P25 Power vs. Time

## Group

Read commands

## Syntax

```
READ:P25:PVTime:TRACe:XY?
```

## Related Commands

```
READ:P25:PVTime:TRACe:X?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:P25:PVTIME:TRACE:XY?` might return #43204 (3204-byte data), which represents the horizontal value (time in seconds) and vertical value (power) pair at the nth data point.

## READ:P25:SUMMARY:LIMIT: SRA (Query Only)

Returns Limit of Symbol Rate Accuracy measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:LIMIT: SRA
```

## Arguments

None

## Returns

might return PASS, FAIL or NA.

---

## Examples

```
READ:P25:SUMMARY:LIMIT: SRA
```

## READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY? (Query Only)

Returns the Modulation Fidelity measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY?
```

### Arguments

None

### Returns

Modulation Fidelity measurement result

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY?
```

## READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY:STATUS (Query Only)

Returns the Modulation Fidelity measurement status.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY:STATUS
```

### Arguments

None

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FIDELITY:STATUS` might return PASS,FAIL or NA.

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY (No Query Form)**

Returns the Frequency Accuracy measurement's result.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Read commands

### Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY`

### Arguments

None

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY`

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY? (Query Only)**

Returns the Operational Frequency Accuracy measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Read commands

### Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY?`

### Arguments

None

### Returns

Frequency Accuracy measurement result

---

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY?
```

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY:STATUS (Query Only)**

Returns the status of Frequency Accuracy measurement status.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY:STATUS
```

### Arguments

None

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ACCURACY:STATUS might return PASS,FAIL or NA.
```

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK**

Returns the Frequency Deviation, negative peak, measurement results.

### Conditions

Measurement view: P25 Summary

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK
```

### Arguments

<integer, integer>

### Returns

Frequency Deviation, negative peak, result

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK<2, 6>` indicates the coordinate value of the Frequency Deviation negative peak.

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK:STATUS**

Returns the Status of Frequency Deviation, negative peak status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK:STATUS`

## Arguments

`<Integer, Integer>`

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:NEGPEAK:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak**

Returns the Frequency Deviation, positive peak, measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPeak`

## Arguments

`<integer, integer>`

## Returns

Frequency Deviation, positive peak, result

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPEAK<-2,-6>` indicates the coordinate value of the Frequency Deviation positive peak.

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPEAK:STATUS**

Returns the status of Frequency Deviation, positive peak status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPEAK:STATUS`

## Arguments

None

## Examples

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:DEVIATION:POSPEAK:STATUS` might return PASS,FAIL or NA.

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR**

Returns the Frequency Error measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR`

## Arguments

None

## Returns

Frequency Error measurement result

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR?
```

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR (Query Only)**

Returns the Frequency Error measurement's result.

## Conditions

Measurement views: P25 Summary

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:FREQUENCY:ERROR
```

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA**

Returns the Symbol Rate Accuracy measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA
```

## Arguments

None



## Returns

Symbol Rate Accuracy measurement result

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA?
```

## **READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUS (Query Only)**

Returns Status of Symbol Rate Accuracy measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUS
```

## Arguments

None

## Returns

Might return PASS, FAIL or NA.

## Examples

```
READ:P25:SUMMARY:MODULATION:MEASUREMENT:SRA:STATUS
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF (Query Only)**

Returns HCPM Pmax-off measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUS? (Query Only)**

Returns HCPM Pmax-off measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUS?
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF:STATUS? might return PASS, FAIL or NA.
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF? (Query Only)**

This command returns the HCPM Pmax-off measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF?
```

## Returns

HCPM Pmax-off measurement results

---

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXOFF?
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON (Query Only)**

Returns HCPM Pmax-on measurement results.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON
```

### Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUS (Query Only)**

Returns Status of HCPM Pmax-on measurement status.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Read commands

### Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUS
```

### Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON:STATUS might return PASS, FAIL or NA.
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON? (Query Only)**

This command returns the HCPM Pmax-on measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON?
```

### **Returns**

HCPM Pmax-on measurement results

### **Examples**

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXON?
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS (Query Only)**

Returns HCPM Pss-max measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS (Query Only)**

Returns HCPM Pss-max measurement status.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS
```

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS? (Query Only)**

Returns the HCPM Pss-max measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS?
```

## Returns

HCPM Pss-on measurement results

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MAXSS?
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM (Query Only)**

Returns HCPM Pss-min measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS? (Query Only)**

Returns Status of HCPM Pss-min measurement results based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS?
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM:STATUS? might return PASS, FAIL or NA.
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM? (Query Only)**

Returns the HCPM Pss-min measurement results.

## Conditions

Measurement view: P25 Summary

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM?
```

## Returns

HCPM Pss-min measurement results

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:MINIMUM?
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSlot (Query Only)**

Returns HCPM Offslot measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSlot
```

## Arguments

None

## Examples

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSlot
```

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSlot:STATUS? (Query Only)**

Returns Status of HCPM Offslot measurement based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

```
READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSlot:STATUS?
```

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:OFFSLOT:STATUS?` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI (Query Only)**

Returns P\_ACP\_HI value which is displayed on P25 Summary display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI`

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:HI` might return -37.963 indicating P\_ACP\_HI value is -37.96 dBm.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:LOW (Query Only)**

Returns P\_ACP\_LOW value which is displayed on P25 Summary display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:LOW`

## Arguments

None



## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:LOW` might return -40.125 indicating P\_ACP\_LOW value is -40.13 dBm.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN (Query Only)**

Returns Min Pk ACPR value which is displayed on P25 Summary Display.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN`

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN` might return 47.96 indicating Min Pk ACPR value is 47.96 dBm

## **READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN:STATUS (Query Only)**

Returns Status of Min Pk ACPR value based on the limit.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN:STATUS`

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:HCPM:PEAK:ACPR:MIN:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:RF:STATUS (Query Only)**

Returns status of RF measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:RF:STATUS`

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:RF:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:POWER:MEASUREMENT:RF (Query Only)**

Returns RF measurement results.

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Read commands

## Syntax

`READ:P25:SUMMARY:POWER:MEASUREMENT:RF`

## Arguments

None

## Examples

`READ:P25:SUMMARY:POWER:MEASUREMENT:RF`

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ON E (Query Only)**

Returns HCPM Time alignment's t\_error\_1 measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ON E:ACQ:COUNT? (Query Only)**

Returns the HCPM Time alignment's t\_error\_1 acquisition count over which the t\_error\_1 averaging is done.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:ACQ:COUNT?
```

### **Returns**

HCPM Time alignment's t\_error\_1 acquisition count over which the t\_error\_1 averaging is done.

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:ACQ:COUNT?
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:STATUS (Query Only)**

Returns status of HCPM Time alignment's t\_error\_1 measurement results based on the limit.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:STATUS
```

### **Arguments**

None

### **Examples**

READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE:STATUS might Return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?(Query Only)**

Returns the HCPM Time alignment t\_error\_1 measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?
```

### **Returns**

HCPM Time alignment t\_error\_1 measurement results.

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ONE?
```

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO (Query Only)**

Returns HCPM Time alignment's t\_error\_0 measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT? (Query Only)**

Returns the HCPM Time alignment's t\_error\_0 acquisition count over which the t\_error\_0 averaging is done.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT?
```

### **Returns**

HCPM Time alignment's t\_error\_0 acquisition count.

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:ACQ:COUNT?
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:STATUS (Query Only)**

Returns status of HCPM Time alignment's t\_error\_0 measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:STATUS`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO? (Query Only)**

Returns the HCPM Time alignment's t\_error\_0 measurement results over which the t\_error\_0 averaging is done.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO?`

### **Returns**

HCPM Time alignment's t\_error\_0 measurement results over which the t\_error\_0 averaging is done.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TERROR:ZERO?`

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync (Query Only)**

Returns HCPM Time alignment's t\_obsync measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync:ACQ:COUNT (Query Only)**

Returns HCPM Time alignment's t\_obsync acquisition counts.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync:ACQ:COUNT
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync:ACQ:COUNT
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT (Query Only)**

Returns Average Transmitter Power Attack acquisition count.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync? (Query Only)**

Returns the HCPM Time alignment's t\_obsync measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync?`

### **Returns**

HCPM Time alignment's t\_obsync measurement results.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:HCPM:TIME:ALIGNMENT:TOBSync?`



---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT (Query Only)**

Returns Average Transmitter Power Attack acquisition count.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT
```

### **Arguments**

None

### **Examples**

READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT: BI (Query Only)**

Returns Average Transmitter Power Attack, busy/idle measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI:STATUS (Query Only)**

Returns status of Average Transmitter Power Encoder Attack Time measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI:STATUS`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:BI:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI?(Query Only)**

Returns the Average Transmitter Power Attack time busy/idle acquisition count over which the Phase 1 Attack time averaging is done.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI?`

### **Returns**

Average Transmitter Power Attack, busy/idle, measurement results.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ACQ:COUNT:BI?`

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS (Query Only)**

Returns Status of Average Transmitter Encoder Attack Time measurement results based on the limit.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER? (Query Only)**

Returns the Average Transmitter Encoder Attack Time measurement results

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER?`

### **Returns**

Average Transmitter Encoder Attack Time measurement results.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:ENCODER?`

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER (Query Only)**

Returns Average Transmitter Power Attack Time measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI (Query Only)**

Returns Average Transmitter Power Attack Time, busy/idle measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI
```

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI:STATUS (Query Only)**

Returns Status of Average Transmitter Power Attack Time, busy/idle measurement results based on the Limit.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI:STATUS`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI? (Query Only)**

Returns the Average Transmitter Power Attack Time, busy/idle measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI?`

### **Returns**

Average Transmitter Power Attack Time, busy/idle, measurement results.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:BI?`

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:STATUS (Query Only)**

Returns Status of Average Transmitter Power Attack Time measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:STATUS`

### **Arguments**

None

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER:STATUS` might return PASS, FAIL or NA.

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER? (Query Only)**

Returns the Average Transmitter Power Attack Time measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER?`

### **Returns**

Average Transmitter Power Attack Time measurement results.

### **Examples**

`READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:ATTACK:TIME:POWER?`

---

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY (Query Only)**

Returns Throughput delay measurement results.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY:ACQ:COUNT (Query Only)**

Returns Throughput delay acquisition counts.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY:ACQ:COUNT
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAY:ACQ:COUNT
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:ACQ:COUNT?(Query Only)**

Returns the Throughput delay acquisition counts over which the Throughput delay averaging is done.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:ACQ:COUNT?
```

### **Returns**

Throughput delay acquisition counts

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:ACQ:COUNT?
```

## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:STATUS (Query Only)**

Returns Status of Throughput delay measurement results based on the limit.

### **Conditions**

Measurement views: P25 Summary

This command requires P25 Measurements.

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:STATUS
```

### **Arguments**

None

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy:STATUS might return PASS, FAIL or NA.
```



## **READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy? (Query Only)**

Returns the Throughput delay measurement results.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Read commands

### **Syntax**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy?
```

### **Returns**

Throughput delay measurement results.

### **Examples**

```
READ:P25:SUMMARY:TRIGGER:MEASUREMENT:PHASE1:THRUput:DELAy?
```

## **READ:PERRor[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the Phase error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

### **Group**

Read commands

### **Syntax**

```
READ:PERRor[:MEASview<y>]:FERRor?
```

### **Arguments**

None

### **Returns**

<freq\_error>::<NRf> is the frequency error in Hz.

## Examples

`READ:PERRor:MEASview2:FERRor?` might return `-10.7E+3`, indicating the frequency error is `-10.7` kHz for measurement view 2.

## **READ:PERRor[:MEASview<y>]:PEAK? (Query only)**

Returns the peak value in the Phase error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Read commands

## Syntax

`READ:PERRor[:MEASview<y>]:PEAK?`

## Related Commands

[\*READ:PERRor\[:MEASview<y>\]:PINdex?\*](#)

## Arguments

None

## Returns

`<peak>::<NRf>` is the peak phase error in percent (%).

## Examples

`READ:PERRor:MEASview2:PEAK?` might return `1.57`, indicating the peak phase error value is `1.57%` for measurement view 2.

## **READ:PERRor[:MEASview<y>]:PINdex? (Query only)**

Returns the time at the phase error peak.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Read commands

## Syntax

```
READ:PERRor[:MEASview<y>]:PINdex?
```

## Related Commands

[READ:PERRor\[:MEASview<y>\]:PEAK?](#)

## Arguments

None

## Returns

`<peak_time>::<NRf>` is the time at the phase error peak in symbol number. The unit can be changed by `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`READ:PERRor:MEASview2:PINdex?` might return 68.000, indicating that the phase error peak is at symbol #68 for measurement view 2.

## READ:PERRor[:MEASview<y>]:RMS (Query only)

Returns the RMS (Root-Mean-Square) value in the Phase error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Read commands

## Syntax

```
READ:PERRor[:MEASview<y>]:RMS?
```

## Arguments

None

## Returns

`<rms>::<NRf>` is the RMS phase error in percent (%).

## Examples

`READ:PERRor:MEASview2:RMS?` might return 0.382, indicating the RMS PERRor value is 0.382% for measurement view 2.

## **READ:PERRor[:MEASview<y>]:TRACe? (Query only)**

Returns the Phase error versus Time trace data.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Read commands

## Syntax

`READ:PERRor[:MEASview<y>]:TRACe?`

## Arguments

None

## Returns

`#<num_digit><num_byte><data(1)><data(2)>...<data(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data(n)>` is the phase error versus Time trace data for the point n in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:PERRor:MEASview2:TRACe?` might return `#42036xxxx...` (2036-byte data) for the Phase error versus Time measview 2.

## **READ:PHVTime[:MEASview<y>]? (Query Only)**

Returns the Phase versus Time trace data.

## Conditions

Measurement views: Phase versus Time

## Group

Read commands

## Syntax

```
READ:PHVTime[:MEASview<y>]?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the phase in degrees at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:PHVTIME:MEASview1? might return #3160xxx... (160-byte data) for the Phase versus Time trace.

## READ:PHVTime[:MEASview<y>]:MAXimum? (Query Only)

Returns the maximum value in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Read commands

## Syntax

```
READ:PHVTime[:MEASview<y>]:MAXimum?
```

## Related Commands

```
READ:PHVTime[:MEASview<y>]:MAXLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max> : : <NRf> is the maximum phase in degrees.

## Examples

READ:PHVTIME:MEASview1:MAXIMUM? might return 153.8, indicating the maximum phase is 153.8 °.

## READ:PHVTime[:MEASview<y>]:MAXLocation? (Query Only)

Returns the time at which the phase is maximum.

## Conditions

Measurement views: Phase versus Time

## Group

Read commands

## Syntax

READ:PHVTime[:MEASview<y>]:MAXLocation?

## Related Commands

READ:PHVTime[:MEASview<y>]:MAXimum?

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<max\_time>::<NRf> is the time in seconds at which the phase is maximum.

## Examples

READ:PHVTIME:MEASview1:MAXLOCATION? might return 175.3E-9, indicating the I or Q level is maximum at 175.3 ns.

## READ:PHVTime[:MEASview<y>]:MINimum? (Query Only)

Returns the minimum value in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Read commands

## Syntax

```
READ:PHVTime[:MEASview<y>]:MINimum?
```

## Related Commands

```
READ:PHVTime[:MEASview<y>]:MINLocation?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min>::<NRf> is the minimum phase in degrees.

## Examples

```
READ:PHVTIME:MEASview1:MINIMUM? might return -176.3, indicating the minimum phase is -176.3 °.
```

## READ:PHVTime[:MEASview<y>]:MINLocation? (Query Only)

Returns the time at which the phase is minimum.

## Conditions

Measurement views: Phase versus Time

## Group

Read commands

## Syntax

```
READ:PHVTime[:MEASview<y>]:MINLocation?
```

## Related Commands

```
READ:PHVTime[:MEASview<y>]:MINimum?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

<min\_time>::<NRf> is the time in seconds at which the phase is minimum.

## Examples

```
READ:PHVTIME:MEASview1:MINLOCATION? might return 450.7E-9, indicating the phase is minimum at 450.7 ns.
```

## **READ:PHVTime[:MEASview<y>]:RESult? (Query Only)**

Returns the Phase versus Time measurement results.

### **Conditions**

Measurement views: Phase versus Time

### **Group**

Read commands

### **Syntax**

```
READ:PHVTime[:MEASview<y>]:RESult?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Returns**

```
<max>,<max_time>,<min>,<min_time>
```

Where

<max>::<NRf> is the maximum phase in degrees.

<max\_time>::<NRf> is the time in seconds at which the phase is maximum.

<min>::<NRf> is the minimum phase in degrees.

<min\_time>::<NRf> is the time in seconds at which the phase is minimum.

### **Examples**

READ:PHVTIME:MEASview1:RESULT? might return 153.8,175.3E-9,-176.3,450.7E-9, indicating the maximum phase is 153.8 ° at 175.3 ns and the minimum phase is -176.3 ° at 450.7 ns.

## **READ:PM:PNEGative? (Query Only)**

Returns the negative peak phase deviation (-Pk) in the PM measurement.

### **Conditions**

Measurement views: PM

### **Group**

Read commands

### **Syntax**

```
READ:PM:PNEGative?
```



## Arguments

None

## Returns

`<-Pk>::<Nrf>` is the negative peak phase deviation in degrees.

## Examples

`READ:PM:PNEGATIVE?` might return `-23.42`, indicating the positive peak phase deviation is `-23.42 °`.

## READ:PM:PPOSitive? (Query Only)

Returns the positive peak phase deviation (+Pk) in the PM measurement.

## Conditions

Measurement views: PM

## Group

Read commands

## Syntax

`READ:PM:PPOSitive?`

## Arguments

None

## Returns

`<+Pk>::<Nrf>` is the positive peak phase deviation in degrees.

## Examples

`READ:PM:PPOSITIVE?` might return `26.87`, indicating the positive peak phase deviation is `26.87 °`.

## READ:PM:PTPeak? (Query Only)

Returns the peak-peak phase deviation (Pk-Pk) in the PM measurement.

## Conditions

Measurement views: PM

## Group

Read commands

## Syntax

`READ:PM:PTPeak?`

## Arguments

None

## Returns

<Pk-Pk> : : <NRf> is the peak-peak phase deviation in degrees.

## Examples

READ:PM:PTPEAK? might return 46.34, indicating the peak-peak phase deviation is 46.34 °.

## READ:PM:RESult? (Query Only)

Returns the PM measurement results.

## Conditions

Measurement views: PM

## Group

Read commands

## Syntax

READ:PM:RESult?

## Arguments

None

## Returns

<+Pk>, <-Pk>, <RMS>, <Pk-Pk>

Where

<+Pk> : : <NRf> is the positive peak phase deviation in degrees.

<-Pk> : : <NRf> is the negative peak phase deviation in degrees.

<RMS> : : <NRf> is the RMS phase deviation in degrees.

<Pk-Pk> : : <NRf> is the peak-peak phase deviation in degrees.

## Examples

READ:PM:RESULT? might return 22.89,-23.45,15.12,46.34.

## READ:PM:RMS? (Query Only)

Returns the RMS phase deviation in the PM measurement.

## Conditions

Measurement views: PM

## Group

Read commands

## Syntax

READ:PM:RMS?

## Arguments

None

## Returns

<RMS>::<NRf> is the RMS phase deviation in degrees.

## Examples

READ:PM:RMS? might return 15.12, indicating the RMS frequency deviation is 15.12 °.

## READ:PNOise:ALL? (Query Only)

Returns all results of the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

READ:PNOise:ALL?

## Arguments

None

## Returns

<Cpower>,<Ferror>,<Pnoise>,<Tjitter>,<Rjitter>,<RFM>

Where

<Cpower>::<NRf> is the carrier power in dBm.

<Ferror>::<NRf> is the frequency error in Hz.

<Pnoise>::<NRf> is the RMS phase noise in degrees.

<Tjitter>::<NRf> is the total jitter in seconds.

<Rjitter>::<NRf> is the random jitter in seconds.

<RFM>::<NRf> is the residual FMin Hz.

## Examples

`READ:PNOISE:ALL?` might return `-9.455,1.235E+6,51.43,2.312E-9,4.178E-9,14.58`, indicating

Carrier power: -9.455 dBm,

Frequency error: 1.235 MHz,

RMS phase noise: 51.43 °,

Total jitter: 2.312 ns,

Random jitter: 4.178 ns, and

Residual FM: 14.58 Hz.

## **READ:PNOise:CARRier:FERRor? (Query Only)**

Returns the carrier frequency error in the phase noise measurement.

### Conditions

Measurement views: Phase noise

### Group

Read commands

### Syntax

`READ:PNOise:CARRier:FERRor?`

### Arguments

None

### Returns

`<NRf>` Carrier frequency error in Hz.

## Examples

`READ:PNOISE:CARRIER:FERROR?` might return `1.235E+6`, indicating that the carrier frequency error is 1.235 MHz.

## **READ:PNOise:CARRier:POWer? (Query Only)**

Returns the carrier power in the phase noise measurement.

### Conditions

Measurement views: Phase noise

### Group

Read commands

## Syntax

READ:PNOise:CARRier:POWer?

## Arguments

None

## Returns

<NRf> Carrier power in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:PNOISE:CARRIER:POWER? might return -9.455, indicating that the carrier power is -9.455 dBm.

## READ:PNOise:JITTer? (Query Only)

Returns the jitter in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

READ:PNOise:JITTer?

## Arguments

None

## Returns

<NRf> Jitter in seconds.

## Examples

READ:PNOISE:JITTER? might return 2.312E-9, indicating that the jitter is 2.312 ns.

## READ:PNOise:RESidual:FM? (Query Only)

Returns the residual FM in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

READ:PNOise:RESidual:FM?

## Arguments

None

## Returns

<NRf> Residual FM in Hz.

## Examples

READ:PNOISE:RESIDUAL:FM? might return 14.58, indicating that the residual FM is 14.58 Hz.

## READ:PNOise:RMS:PNOise? (Query Only)

Returns the RMS phase noise in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

READ:PNOise:RMS:PNOise?

## Arguments

None

## Returns

<NRf> RMS phase noise in degrees.

## Examples

READ:PNOISE:RMS:PNOISE? might return 51.43, indicating that the RMS phase noise is 51.43 °.

## READ:PNOise:SPECTrum<x>:X? (Query Only)

Returns the frequencies of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

```
READ:PNOise:SPECTrum<x>:X?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the frequency (Hz) at the nth point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

```
READ:PNOISE:SPECTRUM1:X? might return #516020xxxx... (16020-byte data) for the frequencies of Trace 1.
```

## READ:PNOise:SPECTrum<x>[:Y]? (Query Only)

Returns the phase noise values of the specified trace.

The parameter <x> = 1 and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

```
READ:PNOise:SPECTrum<x>[:Y]?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><y(1)><y(2)>...<y(n)>
```

Where  $\langle \text{num\_digit} \rangle$  is the number of digits in  $\langle \text{num\_byte} \rangle$ .

$\langle \text{num\_byte} \rangle$  is the number of bytes of data that follow.

$\langle y(n) \rangle$  is the phase noise (dBc/Hz) at the  $n^{\text{th}}$  data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:PNOISE:SPECTRUM1:Y might return #516020xxxx... (16020-byte data) for the phase noise values of Trace 1.

## READ:PNOise:SPECTrum<x>:XY? (Query Only)

Returns the frequency and phase noise pairs of the specified trace.

The parameter  $\langle x \rangle = 1$  and 2, representing Trace 1 and Trace 2, respectively.

## Conditions

Measurement views: Phase noise

## Group

Read commands

## Syntax

READ:PNOise:SPECTrum<x>:XY?

## Arguments

None

## Returns

# $\langle \text{num\_digit} \rangle \langle \text{num\_byte} \rangle \langle x(1) \rangle \langle y(1) \rangle \langle x(2) \rangle \langle y(2) \rangle \dots \langle x(n) \rangle \langle y(n) \rangle$

Where

$\langle \text{num\_digit} \rangle$  is the number of digits in  $\langle \text{num\_byte} \rangle$ .

$\langle \text{num\_byte} \rangle$  is the number of bytes of data that follow.

$\langle x(n) \rangle \langle y(n) \rangle$  is the frequency (Hz) and phase noise (dBc/Hz) pair at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:PNOISE:SPECTRUM1:XY? might return #516020xxxx... (16020-byte data) for the frequency and phase noise pairs of the Trace 1.

## READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer? (Query Only)

Returns the outliers value exceeding the upper limit in pulse cumulative histogram.



## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Read commands

## Syntax

```
READ:PULSE:CUMulative:HISTogram[:MEASview<y>]:OUTLier:HIGHer?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Number of outliers above higher limit

## Examples

READ:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:OUTLIER:HIGHER? might return 0.000, which is the value for the outliers that have exceeded the upper limit in the pulse cumulative histogram.

## READ:PULSE:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer? (Query Only)

Returns the outliers value below the lower limit in pulse cumulative histogram.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Read commands

## Syntax

```
READ:PULSE:CUMulative:HISTogram[:MEASview<y>]:OUTLier:LOWer?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Number of outliers below lower limit

## Examples

`READ:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:OUTLIER:LOWER?` might return 0.000, which is the value for the outliers below the lower limit in the pulse cumulative histogram.

## **READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:COUNT? (Query Only)**

Returns the current events and status conditions for the Cumulative Histogram display.

### Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

### Group

Read commands

### Syntax

`READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:EVENTs?`

### Arguments

None

### Returns

`<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}`

Where `<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description of the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

`READ:PULSe:CUMulative:HISTogram:MEASview1:EVENTS` might return 0,"No events to report".

## **READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:TOTal? (Query Only)**

Returns the total number of pulses in the pulse cumulative histogram.

### Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

### Group

Read commands

### Syntax

`READ:PULSe:CUMulative:HISTogram[:MEASview<y>]:PULse:TOTal?`

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf> Total number of pulses

## Examples

READ:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:PULSE:TOTAL? might return 16 for total number pulses in the pulse cumulative histogram.

## READ:PULSe:CUMulative:STATistics[:MEASview<y>] (Query Only)

Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average and Standard Deviation values for the selected measurement in pulse cumulative statistics.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Read commands

## Syntax

READ:PULSe:CUMulative:STATistics[:MEASview<y>]? <arg1><arg2>

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

arg1 = {AVERAge| PPOWer| ATX| WIDTH| RISE| FALL| RINTerval| RRATe|DUTPct| DUTRatio| RIPPlE| RIPDb| DROOp| DRODb| OVERshoot| OVEDb|PPPHase| PPFRequency| RMSFreqerror| MFReqerror| RMSPherror| MPHerror|FRDeviation| FDELta| PHDeviation| IRAMplitude| IRTIME| PPPD| PPFd| FABS}

<arg2> = {TPCOunt | MAX | MAXTs | MIN | MINTs | PK2PK | AVG | SD | ALL}

## Returns

<puls\_cnt> | <max> | <max\_timstamp> | <min> | <min\_timstamp> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlta\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>

where

<avg\_on>::<NRf> is the average power transmitted during pulse on

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcnt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs. Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlta\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe

## Examples

READ:PULSe:CUMulative:STATistics:MEASview1? AVERage, MAX might return -1.3594852284, indicating the max value for Average ON is -1.3594852284.

---

## **READ:PULSe:OGRAM[:MEASview<y>]:MAX:FRAME:COUNT (Query Only)**

Returns the frame count value in the Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Read commands

### **Syntax**

```
READ:PULSe:OGRAM[:MEASview<y>]:MAX:FRAME:COUNT?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Returns**

<NRf>

### **Examples**

READ:PULSe:OGRAM:MEASview1:MAX:FRAME:COUNT? might return 20, indicating the total number of fast frames in the acquisition of Pulse-Ogram is 20.

## **READ:PULSe:OGRAM[:MEASview<y>]:RBW (Query Only)**

Returns the resolution bandwidth value only when Frequency vs Time display is opened in the Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Read commands

### **Syntax**

```
READ:PULSe:OGRAM[:MEASview<y>]:RBW?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Returns**

<NRf>

## Examples

READ:PULSe:OGRAM:MEASview1:RBW? might return 222.9199886322E+3, indicating the RBW value is 222.9199886322E+3 in Pulse-Ogram.

## READ:PULSe[:RESult][:MEASview<y>]:ADVanced (Query Only)

Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in pulse table.

## Conditions

Measurement views: All

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:ADVanced <arg1>,<arg2>
```

## Arguments

arg1 can be one of the following measurements:

```
{AVERage| PPOwer| ATX|WIDTH| RISE| FALL| RINTerval| RRATe|DUTPct| DUTRatio|
RIPple| RIPDb| DROop| DRODb| OVERshoot|OVEDb| PPPHase| PPFRequency| RMSFreqerror|
MFReqerror|RMSPherror|MPHerror| FRDeviation| FDELta| PHDeviation|IRAmplitude|
IRTime| PPPD| PPFd| FABS}
```

arg2 can be one of the following statistics results:

```
{TPCOunt| MAX| MAXTs| MIN| MINTs| PK2PK| AVG| SD| ALL}
```

## Returns

```
<puls_cnt> | <max> | <min> | <pk_pk> | <avg> | <std_dev> | <all>
```

For one of the following selected measurementsWhere

```
<avg_on> | <pk_pwr> | <avg_tx> | <width> | <rise> | <fall> | <rep_int> | <rep_rate> | <duty_factor_pcmt> | <duty_fact_ratio> | <ripple>
| <ripple_db> | <droop> | <droop_db> | <ovrshoot> | <ovrshoot_db> | <pp_phase> | <pp_freq> | <rms_phase_err> | <rms_freq_err> |
<p2p_phase> | <p2p_freq> | <pk_phase_err> | <pk_freq_err> | <abs_freq> | <freq_dev> | <phase_dev> | <dlt_freq> | <imp_resp_amp> |
<imp_resp_time>|<time> for the given pulse number
```

where<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlta\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe

<time>::<NRf> is the dtme in seconds relative to the time reference point in the first acquisition record in the data set.

## Examples

READ:PULSe:RESult:MEASview2:ADVanced AVERage, MAX might return -1.3594852284 for Average ON value in pulse cumulative statistics

## READ:PULSe[:RESult][[:MEASview<y>]:ADVanced:RANGe (Query only)

Queries the Pulse Count, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the select measurement in the Pulse Table display.

## Conditions

Measurement views: Pulse Table

## Group

Fetch commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:ADVanced:RANGe <arg1><arg2><arg3>
```

## Arguments

<arg1>={Start\_Pulse\_number}

<arg2>={End\_Pulse\_number}

<arg3>= {AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct | DUTRatio | RIPple | RIPDb | DROop | DRODb | OVERshoot | OVEDb | PPPhase | PPFrequency | RMSFreqerror | MFReqerror | RMSPherror | MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude|TIME | IRTIME | PPPD| PPFd | FABS}

## Returns

<puls\_cnt> | <max> | <max\_timstamp> | <min> | <min\_timstamp> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlta\_freq> | <imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number.

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on.

<avg\_tx>:: <NRf> is the average power transmitted.

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB or -6 dB level (50%) of the user selected 100% level.

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% or (20% to 80%) of the user selected 100% level.

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% or (80% to 20%) of the user selected 100% level.

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge.

<rep\_rate> ::<NRf> is the inverse of repetition interval.

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage.

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period.

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top.

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB.

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time.

<droop\_db>::<NRf> is the Droop measurement expressed in dB.

<ovrshoot>:: <NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge.

<ovrshoot\_db>:: <NRf> is the Overshoot measurement expressed in dB.



<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window.

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse.

<rms\_phase\_err>::<NRf> is the RMS average of the Phase versus Time trace, computed over the Measurement Time.

<rms\_freq\_err>::<NRf> is the RMS average of the Freq error versus Time trace.

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse.

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse.

<pk\_phase\_err>::<NRf> is the maximum phase error.

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point.

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time.

<phase\_dev>::<NRf> is the Phase deviation difference between the maximum and minimum Phase values measured during the On time of a pulse.

<dlt\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency.

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe.

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe.

<time>::<NRf> is the dtme in seconds relative to the time reference point in the first acquisition record in the data set.

## Example

```
READ:PULSE[:RESULT]:MEASview1:ADVANCED:RANGE 2048,2060,AVERAge might return
-3,23.8467659801E-3,23.8566566259E-3,23.8667335361E-3,23.8771531731E-3,23.8866508007
E-3,23.8966420293E-3,23.9066667855E-3,23.9170398563E-3,23.9367540926E-3,23.94679561
26E-3,23.9567346871E-3,23.9670332521E-3,23.9866096526E-3,2, indicating the average ON value for the
pulse number from 2048 to 2060.
```

## READ:PULSe:STATistics[:MEASview<y>]:ADVanced (Query Only)

Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in Pulse Statistics display.

## Conditions

Measurement views: Pulse Statistics

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:ADVanced <arg1>,<arg2>
READ:PULSe:STATistics[:MEASview<y>]:AD;Vanced?
```

## Arguments

<arg1>= {AVERAge | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval |

RRATe | DUTPct | DUTRatio |RIPPlE | RIPDb | DROop | DRODb | OVERshoot |  
 OVEDb | PPPHase | PPFRequency | RMSFreqerror | MFReqerror | RMSPherror  
 |MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude|TIME | IRTIME |  
 PPPD| PPFd | FABS}  
 <arg2>= {TPCOunt |MAX |MIN |PK2PK |AVG |SD |ALL}

## Returns

<puls\_cnt> | <max> | <min> | <pk\_pk> | <avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> |  
 <rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> |  
 <droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> |  
 <rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err>  
 | <pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlt\_freq> |  
 <imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB  
 /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20%  
 to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80%  
 to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed  
 as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and  
 the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on

the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs. Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlt\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe

<time>::<NRf> is the dtime in seconds relative to the time reference point in the first acquisition record in the data set

## Examples

READ:PULSE:STATISTICS:MEASview1:ADVANCED RISE, MAX might return 7.9786177665E-9,"-", indicating the maximum rise value is 7.9786177665E-9.

## READ:PULSe[:RESult][[:MEASview<y>]:ATX (Query Only)

Returns the average transmitted power in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][[:MEASview<y>]:ATX?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<ATX(1)>,< ATX(2)>,...<ATX(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ATX(n)>::<NRf> is the average transmitted power for the pulse with the number of [first\_pulse\_num + n - 1] in dBm.

The unit can be changed to watts by the [SENSe]:POWER:UNITs command.

This measurement gives valid results only for power units ( Watts, dBm, dBuW, dBW).

## Examples

```
READ:PULSe:RESult:MEASview2:ATX? might return 0,-18.57,-18.73,-18.20, -18.53 for Pulse 0 to 3.
```

## **READ:PULSe[:RESult][[:MEASview<y>]:AVERage (Query Only)**

Returns the average on power in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][[:MEASview<y>]:AVERage?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<avg(1)>,< avg(2)>,...<avg(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

`<avg (n) > :: <NRf>` is the average on power for the pulse with the number of `[first_pulse_num + n - 1]` in dBm.

The unit can be changed by the `[SENSE] :POWER:UNITS` command.

This measurement gives valid results only for power units ( Watts, dBm, dBuW, dBW).

## Examples

`READ:PULSe:RESult:MEASview2:AVERage?` might return 0,-2.354,-2.368,-2.343,-2.358 for Pulse 0 to 3.

## **READ:PULSe[:RESult][:MEASview<y>]:DRODb (Query Only)**

Returns the droop in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

`READ:PULSe[:RESult][:MEASview<y>]:DRODb?`

## Related Commands

`READ:PULSe[:RESult][:MEASview<y>]:DROop?`

## Arguments

None

## Returns

`<first_pulse_num>,<droop (1)>,<droop (2)>,...<droop (n)>`

Where

`<first_pulse_num> :: <NR1>` is the first pulse number.

`<droop (n) > :: <NRf>` is the droopdB for the pulse number `[first_pulse_num + n - 1]` in percent (%) of watts or volts.

## Examples

`READ:PULSe:RESult:MEASview2:DRODb?` might return 1,-0.4,0.32,0.2,-0.1 for Pulse 1 to 4.

## **READ:PULSe[:RESult][:MEASview<y>]:DROop (Query Only)**

Returns the average on power in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:DROop?
```

## Related Commands

```
READ:PULSe[:RESult][:MEASview<y>]:DRODb?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<droop(1)>,<droop(2)>,...<droop(n)>
```

Where

```
<first_pulse_num>::<NR1> is the first pulse number.
```

```
<droop(n)>::<NRf> is the droop for the pulse number [first_pulse_num + n - 1] in percent (%) of watts or volts.
```

## Examples

```
READ:PULSe:RESult:MEASview2:DROop? might return 0,-270.9E-3,-193.0E-3,-242.7E-3,-177.5E-3 for Pulse 0 to 3.
```

## READ:PULSe[:RESult][:MEASview<y>]:DUTPct (Query Only)

Returns the duty factor (%) in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:DUTPct?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<duty_pct(1)>,<duty_pct(2)>,...<duty_pct(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<duty\_pct (n)>::<NRf> is the duty factor for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%).

## Examples

READ:PULSe:RESult:MEASview2:DUTPct? might return 0,28.94,28.96,29.00,29.01 for Pulse 0 to 3.

## READ:PULSe[:RESult][[:MEASview<y>]:DUTRatio (Query Only)

Returns the duty factor (ratio) in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Read commands

### Syntax

READ:PULSe[:RESult][[:MEASview<y>]:DUTRatio?

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<first\_pulse\_num>,<duty\_ratio(1)>,<duty\_ratio(2)>,... <duty\_ratio(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<duty\_ratio(n)>::<NRf> is the duty factor for the pulse with the number of [first\_pulse\_num + n - 1] (no unit).

## Examples

READ:PULSe:RESult:MEASview2:DUTRatio? might return 0,289.4E-3,289.6E-3, 290.0E-3,290.1E-3 for Pulse 0 to 3.

## READ:PULSe[:RESult][[:MEASview<y>]:FABS (Query Only)

Returns the computed absolute frequency from the carrier in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:FABS?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<abs_freq(1)>,<abs_freq(2)>,...<abs_freq(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<abs\_freq(n)>::<NR1> is the absolute frequency for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

```
READ:PULSe:RESult:MEASview2:FABS? might return 1.0000000000, 919.8726400000E+6, 1.1664642560E+9,  
1.1866586880E+9, 1.1867939840E+9, 1.1882243840E+9, 933.2595840000E+6, 1.1729175040E+9, 1.1818649600E+9,  
1.1868037120E+9, 1.1832893440E+9, 933.4762880000E+6, 1.1805880320E+9, 1.1844794880E+9, 1.1914785280E+9, 1.1929450240E  
for is the number of the first pulse and the absolute frequency for each pulse.
```

## READ:PULSe[:RESult][[:MEASview<y>]:FALL (Query Only)

Returns the fall time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:FALL?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<fall(1)>,<fall(2)>,...<fall(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<fall(n)>::<NRf> is the fall time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.



## Examples

READ:PULSe:RESult:MEASview2:FALL? might return 0,110.3E-9,90.45E-9, 95.03E-9,111.9E-9 for Pulse 0 to 3.

## READ:PULSe[:RESult][:MEASview<y>]:FDELta (Query Only)

Returns the computed delta frequency from the carrier in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Read commands

### Syntax

READ:PULSe[:RESult][:MEASview<y>]:FDELta?

### Arguments

None

### Returns

<first\_pulse\_num>,<freq\_delta(1)>,<freq\_delta(2)>,...<freq\_delta(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<freq\_delta(n)>::<NRf> is the frequency delta for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:FDELta? might return 1.000000000,163.6692962646,82.8298492432,21.8995475769, 89.2963562012,48.1314239502,144.1430664063,52.5188140869, 19.9404506683,287.8957519531,110.4534988403,-110.1940994263, 73.9322433472,115.7962646484,164.6119384766,-42.2758331299, 223.0057525635,28.680925 for the first pulse and each pulse found.

## READ:PULSe[:RESult][:MEASview<y>]:FRDeviation (Query Only)

Returns the frequency deviation in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Read commands

### Syntax

READ:PULSe[:RESult][:MEASview<y>]:FRDeviation?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,<freq\_dev(1)>,<freq\_dev(2)>,...<freq\_dev(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<freq\_dev(n)>::<NRf> is the frequency deviation for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:FRDeviation? might return 1,740.6E+3,736.5E+3,718.3E+3,672.2E+3 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:IRAmplitude (Query Only)

Returns the pulse impulse response amplitude in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

READ:PULSe[:RESult][:MEASview<y>]:IRAmplitude?

## Returns

<first\_pulse\_num>,<amplitude(1)>,<amplitude(2)>,...<amplitude(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<amplitude(n)>::<NRf> is the sidelobe level, in dB, from the mainlobe for the pulse with the number of [first\_pulse\_num + n - 1].

## Examples

READ:PULSe:RESult:MEASview2:IRAmplitude? might return 1.0000000000,-45.3,-47.5,-50.2 for Pulse 1 through 3.

## READ:PULSe[:RESult][:MEASview<y>]:IRTime (Query Only)

Returns the impulse response time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:IRTime?
```

## Returns

```
<first_pulse_num>,<time(1)>,<time(2)>,...<time(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<time(n)>::<NRf> is the sidelobe offset time from the mainlobe for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

```
READ:PULSe:RESult:MEASview2:IRTime? might return
1.0000000000,806.6708687693E-9,-846.6595318168E-9,-873.3477443457E-9 for Pulse 1 to 3.
```

## READ:PULSe[:RESult][[:MEASview<y>]:MFReqerror (Query Only)

Returns the maximum frequency error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:MFReqerror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<max_freq_err(1)>,<max_freq_err(2)>,...<max_freq_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<max\_freq\_err(n)>::<NRf> is the maximum frequency error for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

```
READ:PULSe:RESult:MEASview2:MFReqerror? might return 1,597.5E+3,675.8E+3,642.8E+3,598.2E+3 for Pulse 1 to 4.
```

## **READ:PULSe[:RESult][:MEASview<y>]:MPHerror (Query Only)**

Returns the maximum phase error in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Read commands

### **Syntax**

```
READ:PULSe[:RESult][:MEASview<y>]:MPHerror?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>,<max_phase_err(1)>,<max_phase_err(2)>,...<max_phase_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<max\_phase\_err(n)>::<NRf> is the maximum phase error for the pulse with the number of [first\_pulse\_num + n - 1] in degrees.

### **Examples**

READ:PULSe:RESult:MEASview2:MPHerror? might return 1,-9.221,-8.413,-11.853,-10.258 for Pulse 1 to 4.

## **READ:PULSe[:RESult][:MEASview<y>]:OVERDb (Query Only)**

Returns the overshoot in dB in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Read commands

### **Syntax**

```
READ:PULSe[:RESult][:MEASview<y>]:OVERDb?
```

### **Related Commands**

```
READ:PULSe[:RESult][:MEASview<y>]:OVERshoot?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <overshoot (1)>, <overshoot (2)>, ... <overshoot (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<overshoot (n)>::<NRf> is the overshoot for the pulse with the number of [first\_pulse\_num + n - 1] in dB.

## Examples

READ:PULSe:RESult:MEASview2:OVERDb? might return 1,1.2,2.3,1.0,0.2 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:OVERshoot (Query Only)

Returns the overshoot in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

READ:PULSe[:RESult][:MEASview<y>]:OVERshoot?

## Related Commands

READ:PULSe[:RESult][:MEASview<y>]:OVERDb?

## Arguments

None

## Returns

<first\_pulse\_num>, <overshoot (1)>, <overshoot (2)>, ... <overshoot (n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<overshoot (n)>::<NRf> is the overshoot for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%) of watts or volts.

## Examples

READ:PULSe:RESult:MEASview2:OVERshoot? might return 1,1.2,2.3,1.0,0.2 for Pulse 1 through 4.

## **READ:PULSe[:RESult][:MEASview<y>]:PHDeviation (Query Only)**

Returns the phase deviation in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Read commands

### **Syntax**

```
READ:PULSe[:RESult][:MEASview<y>]:PHDeviation?
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

```
<first_pulse_num>,<phase_dev(1)>,<phase_dev(2)>,...<phase_dev(n)>
```

Where

```
<first_pulse_num>::<NR1> is the first pulse number.
```

```
<phase_dev(n)>::<NRf> is the phase deviation for the pulse with the number of [first_pulse_num + n - 1] in degrees.
```

### **Examples**

```
READ:PULSe:RESult:MEASview2:PHDeviation? might return 1,11.658,9.640,10.509,8.272 for Pulse 1 to 4.
```

## **READ:PULSe[:RESult][:MEASview<y>]:PPFD (Query Only)**

Returns the computed pulse to pulse frequency difference from the carrier in the pulse table.

### **Conditions**

Measurement views: Pulse table

### **Group**

Read commands

### **Syntax**

```
READ:PULSe[:RESult][:MEASview<y>]:PPFD
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <ppfd(1)>, <ppfd(2)>, .....<ppfd(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ppfd(n)>::<NRf> is the pulse to pulse frequency difference with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:PPFD? might return 1.0000000000,"-", 246.5916160000E+6, 20.1943520000E+6, 135.2960000000E+3, 1.4304160000E+6, -254.9647520000E+6, 239.6579520000E+6, 8.9474560000E+6, 4.9386560000E+6, -3.5143840000E+6, -249.8130240000E+6, 247.1117760000E+6, 3.8915200000E+6, 6.9989120000E+6, 1.4665920000E+6 is the number of the first pulse and the pulse frequency difference for each pulse.

## READ:PULSe[:RESult][:MEASview<y>]:PPFRequency (Query Only)

Returns the pulse-pulse carrier frequency in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

READ:PULSe[:RESult][:MEASview<y>]:PPFRequency?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <pp\_freq(1)>, <pp\_freq(2)>, ...<pp\_freq(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pp\_freq(n)>::<NRf> is the pulse-pulse carrier frequency for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:PPFRequency? might return 0,0.000,1.258E+3,-3.121E+3,1.862E+3 for Pulse 0 to 3.

## READ:PULSe[:RESult][:MEASview<y>]:PPOWer (Query Only)

Returns the peak power in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:PPOWer?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<pk_power(1)>,<pk_power(2)>,...<pk_power(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<pk\_power(n)>::<NRf> is the peak power for the pulse with the number of [first\_pulse\_num + n - 1] in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

This measurement gives valid results only for power units( Watts, dBm, dBuW, dBW).

## Examples

```
READ:PULSe:RESult:MEASview2:PPOWer? might return 0,-2.26,-2.27,-2.23,-2.25 for Pulse 0 to 3.
```

## READ:PULSe[:RESult][:MEASview<y>]:PPPD (Query Only)

Returns the computed pulse to pulse phase difference from the carrier in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:PPPD
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.



## Returns

`<first_pulse_num>, <pppd(1)>, <pppd(2)>, . . . . . <pppd(n)>`

Where

`<first_pulse_num>::<NR1>` is the first pulse number.

`<pppd(n)>::<NRf>` is the pulse to pulse phase difference with the number of  $[\text{first\_pulse\_num} + n - 1]$  in Hz.

## Examples

`READ:PULSe:RESult:MEASview2:PPPD?` might return 1.0000000000,"-", 96.2469482422, 24.9118175507, 5.8206896782, 16.7037734985, -151.8063964844, 100.7584228516, 36.6761512756, -3.8614616394, 12.9028282166, -150.9820556641, 130.5854492188, 9.1259155273, -7.4177660942, 6.0613451004, -142.4224853516 is the number of the first pulse and the pulse to pulse phase difference for each pulse.

## READ:PULSe[:RESult][:MEASview<y>]:PPHase (Query Only)

Returns the pulse-pulse carrier phase in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

`READ:PULSe[:RESult][:MEASview<y>]:PPHase?`

## Arguments

The parameter `<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

`<first_pulse_num>, <pp_phase(1)>, <pp_phase(2)>, . . . <pp_phase(n)>`

Where

`<first_pulse_num>::<NR1>` is the first pulse number.

`<pp_phase(n)>::<NRf>` is the pulse-pulse carrier phase for the pulse with the number of  $[\text{first\_pulse\_num} + n - 1]$  in degrees.

## Examples

`READ:PULSe:RESult:MEASview2:PPHase?` might return 0,0.000,21.66,46.76,57.56 for Pulse 0 to 3.

## READ:PULSe[:RESult][:MEASview<y>]:RINTerval (Query Only)

Returns the repetition interval in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:RINTerval?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<rep_int(1)>,<rep_int(2)>,...<rep_int(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<rep\_int(n)>::<NRf> is the repetition interval for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

```
READ:PULSe:RESult:MEASview2:RINTerval? might return 0,16.03E-6,16.08E-6, 16.07E-6,16.02E-6 for Pulse 0 to 3.
```

## READ:PULSe[:RESult][:MEASview<y>]:RIPDb (Query Only)

Returns the ripple in dB in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:RIPDb?
```

## Related Commands

```
READ:PULSe[:RESult][:MEASview<y>]:RIPPlE?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <ripple(1)>, <ripple(2)>, ... <ripple(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ripple(n)>::<NRf> is the ripple for the pulse with the number of [first\_pulse\_num + n - 1] in dB.

## Examples

READ:PULSe:RESult:MEASview2:RIPDb? might return 1,0.4,0.32,0.2,0.1 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:RIPPlE (Query Only)

Returns the ripple in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

READ:PULSe[:RESult][:MEASview<y>]:RIPPlE?

## Related Commands

READ:PULSe[:RESult][:MEASview<y>]:RIPDb?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>, <ripple(1)>, <ripple(2)>, ... <ripple(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<ripple(n)>::<NRf> is the watt or voltage ripple for the pulse with the number of [first\_pulse\_num + n - 1] in percent (%).

## Examples

READ:PULSe:RESult:MEASview2:RIPPlE? might return 0,106.5E-3,177.6E-3, 148.3E-3,148.5E-3 for Pulse 0 to 3.

## READ:PULSe[:RESult][:MEASview<y>]:RISE (Query Only)

Returns the rise time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:RISE?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<rise(1)>,<rise(2)>,...<rise(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<rise(n)>::<NRf> is the rise time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

```
READ:PULSe:RESult:MEASview2:RISE? might return 0,92.94E-9,115.9E-9, 115.1E-9,97.45E-9 for Pulse 0 to 3.
```

## READ:PULSe[:RESult][:MEASview<y>]:RMSFreqerror (Query Only)

Returns the RMS frequency error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:RMSFreqerror?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<RMS_freq_err(1)>,<RMS_freq_err(2)>,...<RMS_freq_err(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<RMS\_freq\_err(n)>::<NRf> is the RMS frequency error for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:RMSFreqerror? might return 1,51.54E+3, 69.20E+3,64.21E+3,51.02E+3 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:RMSPherror (Query Only)

Returns the RMS phase error in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

READ:PULSe[:RESult][:MEASview<y>]:RMSPherror?

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<first\_pulse\_num>,<RMS\_phase\_err(1)>,<RMS\_phase\_err(2)>,...<RMS\_phase\_err(n)>

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<RMS\_phase\_err(n)>::<NRf> is the RMS phase error for the pulse with the number of [first\_pulse\_num + n - 1] in degrees.

## Examples

READ:PULSe:RESult:MEASview2:RMSPherror? might return 1,908.4E-3,752.8E-3,981.7E-3,886.4E-3 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:RRATe (Query Only)

Returns the repetition rate in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][[:MEASview<y>]:RRATE?
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<rep_rate(1)>,<rep_rate(2)>,...<rep_rate(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<rep\_rate(n)>::<NRf> is the repetition rate for the pulse with the number of [first\_pulse\_num + n - 1] in Hz.

## Examples

READ:PULSe:RESult:MEASview2:RRATE? might return 0,62.50E+3,62.52E+3, 62.51E+3,62.49E+3 for Pulse 0 to 3.

## READ:PULSe[:RESult][[:MEASview<y>]:TIME (Query Only)

Returns the time in the pulse table.

## Conditions

Measurement views: Pulse table

## Group

Read commands

## Syntax

```
READ:PULSe[:RESult][[:MEASview<y>]:TIME
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<first_pulse_num>,<time(1)>,<time(2)>,...<time(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<time(n)>::<NRf> is the time for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

## Examples

READ:PULSe:RESult:MEASview2:TIME? might return 1,7.937E-3,8.436E-3, 6.504E-3,9.876E-3 for Pulse 1 to 4.

## READ:PULSe[:RESult][:MEASview<y>]:WIDTh (Query Only)

Returns the pulse width in the pulse table.

### Conditions

Measurement views: Pulse table

### Group

Read commands

### Syntax

```
READ:PULSe[:RESult][:MEASview<y>]:WIDTh?
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

```
<first_pulse_num>,<width(1)>,<width(2)>,...<width(n)>
```

Where

<first\_pulse\_num>::<NR1> is the first pulse number.

<width(n)>::<NRf> is the pulse width for the pulse with the number of [first\_pulse\_num + n - 1] in seconds.

### Examples

READ:PULSe:RESult:MEASview2:WIDTh? might return 0,4.630E-6,4.632E-6, 4.639E-6,4.642E-6 for Pulse 0 to 3.

## READ:PULSe:STATistics[:MEASview<y>] (Query Only)

Returns the trace data of the pulse statistics measurement selected by the DISPLAY:PULSe:SElect:RESult command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the DISPLAY:PULSe:STATistics:PLOT command before executing this query.

### Conditions

Measurement views: Pulse statistics

### Group

Read commands

### Syntax

```
READ:PULSe:STATistics[:MEASview<y>]?
```

### Arguments

None

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n)> is the amplitude at the nth data point.

The unit is current measurement unit (Plot = Trend or Time Trend), dB (Plot = FFT), count or percent (Plot = Histogram)

4-byte little endian floating-point format specified in IEEE 488.2.

The unit of power is selected by the [SENSe]:POWer:UNITs command.

## Examples

READ:PULSe:STATistics:MEASview1? might return #264xxx... (64-byte data) for the statistics trace of the pulse width measurement when DISPLAY:PULSe:SElect:RESult is set to WIDTH.

## READ:PULSe:STATistics[:MEASview<y>]:ADVanced (Query Only)

Returns the Pulse Count, Max, Max at Pulse number, Min, Min at Pulse number, Peak to Peak, Average and Standard Deviation values for the chosen measurement in pulse statistics display.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:ADVanced? <arg1>,<arg2>

## Arguments

arg1 can be one of the following measurements:

{AVERage| PPOWer| ATX|WIDTH| RISE| FALL| RINTerval| RRATe|DUTPct| DUTRatio| RIPple| RIPDb| DROop| DRODb| OVERshoot|OVEDb| PPPHase| PPFRequency| RMSFReqerror| MFReqerror|RMSPherror|MPHerror| FRDeviation| FDELta| PHDeviation|IRAMplitude| IRTime| PPPD| PPFd| FABS}

arg2 can be one of the following statistics results:

{TPCOunt| MAX| MIN| PK2PK| AVG| SD| ALL}

## Returns

<puls\_cnt>|<max>|<min>|<pk\_pk>|<avg>|<std\_dev>|<all> for one of the following selected measurements.

Where

<avg\_on>|<pk\_pwr>|<avg\_tx>|<width>|<rise>|<fall>|<rep\_int>|<rep\_rate>|<



<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> | <rep\_rate>  
 | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> | <droop> |  
 <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> | <rms\_phase\_err>  
 | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> | <pk\_freq\_err>  
 | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlta\_freq> | <imp\_resp\_amp> |  
 <imp\_resp\_time>|<time>

#### Where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlta\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and highest side lobe

<time>::<NRf> is the dtime in seconds relative to the time reference point in the first acquisition record in the data set.

## Examples

READ:PULSe:STATistics:MEASview2:ADVanced RISE, MAX might return 7.9786177665E-9,"--", indicating the maximum rise value is 7.9786177665E-9.

## READ:PULSe[:RESult]:ADVanced (Query Only)

Returns the PulseCount, Max, Max TimeStamp, Min, Min TimeStamp, Peak to Peak, Average, and Standard Deviation values for the chosen measurement in the Pulse Table display.

## Conditions

Measurement views: All

## Group

Read commands

## Syntax

READ:PULSe[:RESult]:ADVanced? <arg1><arg2>

## Arguments

<arg1>={Pulse\_number}

<arg2>= {AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval |  
RRATe | DUTPct | DUTRatio |RIPPlE | RIPDb | DROop | DRODb | OVERshoot |  
OVEDb | PPPHase | PPFRequency | RMSFreqerror | MFRequerror | RMSPherror  
|MPHerror | FRDeviation | FDELta | PHDeviation | IRAMplitude|TIME | IRTime |  
PPPD| PPFd | FABS}

## Returns

<puls\_cnt> | <max> | <max\_timestamp> | <min> | <min\_timestamp> | <pk\_pk> |  
<avg> | <std\_dev> | <all>

For one of the following selected measurements:

<avg\_on> | <pk\_pwr> | <avg\_tx> | <width> | <rise> | <fall> | <rep\_int> |  
<rep\_rate> | <duty\_factor\_pcmt> | <duty\_fact\_ratio> | <ripple> | <ripple\_db> |  
<droop> | <droop\_db> | <ovrshoot> | <ovrshoot\_db> | <pp\_phase> | <pp\_freq> |  
<rms\_phase\_err> | <rms\_freq\_err> | <p2p\_phase> | <p2p\_freq> | <pk\_phase\_err> |  
<pk\_freq\_err> | <abs\_freq> | <freq\_dev> | <phase\_dev> | <dlt\_freq> |  
<imp\_resp\_amp> | <imp\_resp\_time>|<time> for the given pulse number

where

<avg\_on>::<NRf> is the average power transmitted during pulse on.

<pk\_pwr>::<NRf> is the Maximum power during pulse on

<avg\_tx>:: <NRf> is the average power transmitted

<width>::<NRf> is the time from the rising edge to the falling edge at the -3 dB /-6 dB level (50%) of the user selected 100% level

<rise>::<NRf> is the time required for a signal to rise from 10% to 90% (or 20% to 80%) of the user selected 100% level

<fall> ::<NRf> is the time required for a signal to fall from 90% to 10% (or 80% to 20%) of the user selected 100% level

<rep\_int>::<NRf> is the time from a pulse rising edge to the next pulse rising edge

<rep\_rate> ::<NRf> is the inverse of repetition interval

<duty\_factor\_pcmt>::<NRf> is the ratio of the width to the pulse period, expressed as a percentage

<duty\_fact\_ratio>::<NRf> is the ratio of the pulse width to the pulse period

<ripple>::<NRf> is the Ripple is the peak-to-peak ripple on the pulse top

<ripple\_db>::<NRf> is the Ripple measurement expressed in dB

<droop>::<NRf> is the Droop is the power difference between the beginning and the end of the pulse On time

<droop\_db>::<NRf> is the Droop measurement expressed in dB

<ovrshoot>::<NRf> is the amount by which the signal exceeds the 100% level on the pulse rising edge

<ovrshoot\_db>::<NRf> is the Overshoot measurement expressed in dB

<pp\_phase>::<NRf> is the phase difference between the selected pulse and the previous pulse in the analysis window

<pp\_freq> ::<NRf> is the difference between the frequency of the current pulse and frequency of the previous pulse

<rms\_phase\_err>::<NRf> is the RMS average of the Phase vs Time trace, computed over the Measurement Time

<rms\_freq\_err>::<NRf> is the RMS average of the Freq Error vs.Time trace

<p2p\_phase>::<NRf> is the frequency difference between the current pulse and the first pulse

<p2p\_freq>::<NRf> is the frequency difference between the current pulse and immediate previous pulse

<pk\_phase\_err>::<NRf> is the maximum phase error

<abs\_freq>::<NRf> is the absolute pulse frequency measured at a specific point

<freq\_dev>::<NRf> is the difference between the maximum and minimum

measured values of the signal frequency during the Measurement Time

<phase\_dev>::<NRf> is the Phase Deviation difference between the maximum and minimum Phase values measured during the On time of a pulse

<dlt\_freq>::<NRf> is the difference from the measurement frequency to each pulse frequency

<imp\_resp\_amp>::<NRf> is the difference in dB between the levels of the main lobe and highest side lobe

<imp\_resp\_time>::<NRf> is the difference in time between the main lobe and

highest side lobe <time>::<NRf> is the dtime in seconds relative to the time reference point in the first acquisition record in the data set

## Examples

READ:PULSE[:RESULT]:ADVANCED 1,AVERAge might return -1.4063608208, indicating the average ON value for the pulse number 1 is -1.4063608208.

## READ:PULSe:STATistics[:MEASview<y>]:AVERAge (Query Only)

Returns the average on power in the pulse statistics. This command is valid when

DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:AVERAge

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<avg\_avg>,<avg\_min>,< avg\_max>

Where

<avg\_avg>::<NRf> is the average of the average on power.

<avg\_min>::<NRf> is the minimum of the average on power.

<avg\_max>::<NRf> is the maximum of the average on power.

Unit: dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

`READ:PULSe:STATistics:MEASview2:AVERage` might return -2.35,-2.36,-2.34 for the average on power in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:DRODb (Query Only)**

Returns the droop in dB in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

`READ:PULSe:STATistics[:MEASview<y>]:DRODb`

## Related Commands

`READ:PULSe:STATistics[:MEASview<y>]:DROop?`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`<droop_avg>`, `<droop_min>`, `<droop_max>`

Where

`<droop_avg>`::`<NRf>` is the average droop.

`<droop_min>`::`<NRf>` is the minimum droop.

`<droop_max>`::`<NRf>` is the maximum droop.

Unit: in dB by watts.

## Examples

`READ:PULSe:STATistics:MEASview2:DRODb?` might return 22.67E-3,-613.5E-3, 633.8E-3 for the droop in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:DROop (Query Only)**

`READ:PULSe:STATistics[:MEASview<y>]:DROop`

Returns the droop in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:DROop
```

## Related Commands

```
READ:PULSe:STATistics[:MEASview<y>]:DRODb?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<droop_avg>, <droop_min>, <droop_max>
```

Where

<droop\_avg>::<NRf> is the average droop.

<droop\_min>::<NRf> is the minimum droop.

<droop\_max>::<NRf> is the maximum droop.

Unit: Percent (%) by watts.

## Examples

READ:PULSe:STATistics:MEASview2:DROop might return 22.67E-3,-613.5E-3, 633.8E-3 for the droop in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:DUTPct (Query Only)

Returns the duty factor (%) in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:DUTPct
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<duty\_pct\_avg>, <duty\_pct\_min>, <duty\_pct\_max>

Where

<duty\_pct\_avg>::<Nrf> is the average duty factor.

<duty\_pct\_min>::<Nrf> is the minimum duty factor.

<duty\_pct\_max>::<Nrf> is the maximum duty factor.

Unit: Percent (%).

## Examples

READ:PULSe:STATistics:MEASview2:DUTPct might return 2.437,2.310,2.657 for the duty factor (%) in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:DUTRatio (Query Only)

Returns the duty factor (ratio) in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:DUTRatio

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<duty\_ratio\_avg>, <duty\_ratio\_min>, <duty\_ratio\_max>

Where

<duty\_ratio\_avg>::<Nrf> is the average duty factor.

<duty\_ratio\_min>::<Nrf> is the minimum duty factor.

<duty\_ratio\_max>::<Nrf> is the maximum duty factor.

Unit: None.

## Examples

`READ:PULSe:STATistics:MEASview2:DUTRatio?` might return 24.37E-3,23.11E-3, 26.57E-3 for the duty factor (ratio) in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:FABS (Query Only)**

Returns the absolute frequency in pulse statistics.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:FABS
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<fabs_avg>,<fabs_min>,<fabs_max>
```

Where

<fabs\_avg>::<NRf> is the average of fabs.

<fabs\_min>::<NRf> is the minimum of fabs.

<fabs\_max>::<NRf> is the maximum of fabs.

## Examples

`READ:PULSe:STATistics:MEASview2:FABS` might return 1.1196544000E+9,919.8726400000E+6,1.1929450240E+9 for the absolute frequency in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:FALL (Query Only)**

Returns the fall time in the pulse statistics. This command is valid when

`DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands



## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:FALL
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<fall_avg>, <fall_min>, <fall_max>
```

Where

<fall\_avg>::<NRf> is the average fall time.

<fall\_min>::<NRf> is the minimum fall time.

<fall\_max>::<NRf> is the maximum fall time.

Unit: Seconds.

## Examples

READ:PULSe:STATistics:MEASview2:FALL? might return 70.27E-9,69.62E-9, 71.27E-9 for the fall time in the pulse statistics.

## READ:PULSe:STATistics[:MEASview<y>]:FDELta (Query Only)

Return the delta frequency measurement in the pulse statistics.

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:FDELta
```

## Arguments

None

## Returns

```
<freq_delta_avg>, <freq_delta_min>, <freq_delta_max>
```

Where

<freq\_delta\_avg>::<NRf> is the average frequency deviation.

<freq\_delta\_min>::<NRf> is the minimum frequency deviation.

<freq\_delta\_max>::<NRf> is the maximum frequency deviation.

Unit: Hz.

## Examples

`READ:PULSe:STATistics:MEASview2:FDELta?` might return 77.9785690308,-110.1940994263,287.8957519531 for the frequency delta in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:FRDeviation (Query Only)**

Returns the frequency deviation in the Pulse Statistics measurement . This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:FRDeviation
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

```
<freq_dev_avg>,<freq_dev_min>,<freq_dev_max>
```

Where

<freq\_dev\_avg>::<NRf> is the average frequency deviation.

<freq\_dev\_min>::<NRf> is the minimum frequency deviation.

<freq\_dev\_max>::<NRf> is the maximum frequency deviation.

Unit: Hz.

## Examples

`READ:PULSe:STATistics:MEASview2:FRDeviation?` might return 754.1E+3, 660.5E+3,835.2E+3 for the frequency deviation in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:IRAMplitude (Query Only)**

Returns the Pulse Impulse Response amplitude measurement in the pulse statistics.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:IRAMplitude
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<amp;avg>, <amp;min>, <amp;max>
```

Where

<amp;avg>::<Nrf> is the average of the impulse response amplitude.

<amp;min>::<Nrf> is the minimum of the impulse response amplitude.

<amp;max>::<Nrf> is the maximum of impulse response amplitude.

Unit: dB.

## Examples

READ:PULSe:STATistics:MEASview2:IRAMplitude? might return -45.3,-47.1,-43.2 for the impulse response amplitude in the pulse statistics.

## READ:PULSe:STATistics[:MEASview<y>]:IRTime (Query Only)

Returns the impulse response time in the pulse statistics.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:IRTime
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<avg_avg>, <avg_min>, <avg_max>
```

Where

<avg\_avg>::<Nrf> is the average of the impulse response time.

<avg\_min>::<Nrf> is the minimum of the impulse response time.

<avg\_max>::<Nrf> is the maximum of impulse response time.

Unit: seconds.

## Examples

`READ:PULSe:STATistics:MEASview2:IRTime?` might return -2.35,-2.36,-2.34 for the impulse response time in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:MAX:PULSe:NUMber (Query Only)**

Returns the maximum value at a certain pulse number in the pulse statistics display.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

### Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:MAX:PULSe:NUMber
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<max\_pulse\_no>

Where

<max\_pulse\_no>::<NRf> is the maximum value at a pulse number.

### Examples

`READ:PULSe:STATistics:MEASview2:MAX:PULSe:NUMber` might return 15 for the maximum pulse number in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:MFReqerror (Query Only)**

Returns the maximum frequency error in the pulse statistics measurement. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:MFRqerror
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<max_freq_err_avg>, <max_freq_err_min>, <max_freq_err_max>
```

Where

<max\_freq\_err\_avg>::<NRf> is the average of maximum frequency error.

<max\_freq\_err\_min>::<NRf> is the minimum of maximum frequency error.

<max\_freq\_err\_max>::<NRf> is the maximum of maximum frequency error.

Unit: Hz.

## Examples

READ:PULSe:STATistics:MEASview2:MFRqerror? might return 645.0E+3,555.6E+3,738.8E+3 for the maximum frequency error in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMBER? (Query Only)

Returns the minimum value at a certain pulse number in pulse statistics.

## Conditions

Measurement views: Pulse statistics

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:MIN:PULSe:NUMBER?
```

## Arguments

None

## Returns

```
<min_pulse_no>
```

Where

<min\_pulse\_no>::<NRf> is the minimum value at a pulse number.

## Examples

`READ:PULSE:STATISTICS:MEASview1:MIN:PULSE:NUMBER?` might return 1 for the minimum pulse number in the pulse statistics

## **READ:PULSe:STATistics[:MEASview<y>]:MPHerror (Query Only)**

Returns the maximum phase error in the Pulse Statistics measurement. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

### Syntax

`READ:PULSe:STATistics[:MEASview<y>]:MPHerror`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`<max_phase_err_avg>`, `<max_phase_err_min>`, `<max_phase_err_max>`

Where

`<max_phase_err_avg>::<NRf>` is the average of maximum phase error.

`<max_phase_err_min>::<NRf>` is the minimum of maximum phase error.

`<max_phase_err_max>::<NRf>` is the maximum of maximum phase error.

Unit: Degrees.

### Examples

`READ:PULSe:STATistics:MEASview2:MPHerror?` might return -11.47,-17.18,-7.61 for the maximum phase error in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:OVEdB (Query Only)**

Returns the overshoot measurement result in dB in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND` or `TTRend`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:OVEDb
```

## Related Commands

```
READ:PULSe:STATistics[:MEASview<y>]:OVERshoot
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<overshoot_avg>,<overshoot_min>,<overshoot_max>
```

Where

<overshoot\_avg>::<NRf> is the average overshoot.

<overshoot\_min>::<NRf> is the minimum overshoot.

<overshoot\_max>::<NRf> is the maximum overshoot.

Unit: dB.

## Examples

READ:PULSe:STATistics:MEASview2:OVEDb might return 0.3,0.1,0.5 for the overshoot measurement result in dB in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:OVERshoot (Query Only)

Returns the overshoot in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd or TTRend.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:OVERshoot
```

## Related Commands

```
READ:PULSe:STATistics[:MEASview<y>]:OVEDb
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<overshoot\_avg>, <overshoot\_min>, <overshoot\_max>

Where

<overshoot\_avg>::<NRf> is the average overshoot.

<overshoot\_min>::<NRf> is the minimum overshoot.

<overshoot\_max>::<NRf> is the maximum overshoot.

Unit: Percent (%) as watts or volts.

## Examples

READ:PULSe:STATistics:MEASview2:OVERshoot might return 0.3,0.1,0.5 for the overshoot in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:PHDeviation (Query Only)

Returns the phase deviation in the Pulse Statistics measurement . This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:PHDeviation

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<phase\_dev\_avg>, <phase\_dev\_min>, <phase\_dev\_max>

Where

<phase\_dev\_avg>::<NRf> is the average phase deviation.

<phase\_dev\_min>::<NRf> is the minimum phase deviation.

<phase\_dev\_max>::<NRf> is the maximum phase deviation.

Unit: Degrees.



## Examples

`READ:PULSe:STATistics:MEASview2:PHDeviation` might return 11.678,7.694, 17.374 for the phase deviation in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:PPFD (Query Only)**

Returns the pulse-pulse frequency difference in the Pulse Statistics measurement display.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

### Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:PPFD
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

```
<ppfd_avg>,<ppfd_min>,<ppfd_max>
```

Where

<ppfd\_avg>::<NRf> is the average of ppfd.

<ppfd\_min>::<NRf> is the minimum of ppfd.

<ppfd\_max>::<NRf> is the maximum of ppfd.

## Examples

`READ:PULSe:STATistics:MEASview2:PPFD` might return 365.4186562500E+3,-267.5911040000E+6,247.1117760000E+6 for the pulse-pulse frequency difference in the pulse statistics.

## **READ:PULSe:STATistics[:MEASview<y>]:PPFrequency (Query Only)**

Returns the pulse-pulse carrier frequency in the pulse statistics measurement. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:PPFrequency
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<pp_freq_avg>,<pp_freq_min>,<pp_freq_max>
```

Where

<pp\_freq\_avg>::<NRf> is the average pulse-pulse carrier frequency.

<pp\_freq\_min>::<NRf> is the minimum pulse-pulse carrier frequency.

<pp\_freq\_max>::<NRf> is the maximum pulse-pulse carrier frequency.

Unit: Hz.

## Examples

READ:PULSe:STATistics:MEASview2:PPFrequency? might return 1.135E+3,311.3E+3,-262.8E+3 for the pulse-pulse carrier frequency in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:PPOWER (Query Only)

Returns the peak power in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:PPOWER
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<pk_power_avg>,<pk_power_min>,<pk_power_max>
```

Where

<pk\_power\_avg>::<NRf> is the average peak power.

<pk\_power\_min>::<NRf> is the minimum peak power.

`<pk_power_max>::<NRf>` is the maximum peak power.

Unit: dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

## Examples

`READ:PULSe:STATistics:MEASview2:PPOwer?` might return -2.273,-2.313,-2.235 for the peak power in the pulse statistics measurement .

## READ:PULSe:STATistics[:MEASview<y>]:PPPD (Query Only)

Returns the overshoot measurement result in dB in the pulse statistics measurement. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND` or `TTRend`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

`READ:PULSe:STATistics[:MEASview<y>]:PPPD`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`<pppd_avg>`, `<pppd_min>`, `<pppd_max>`

Where

`<pppd_avg>::<NRf>` is the average of `pppd`.

`<pppd_min>::<NRf>` is the minimum of `pppd`.

`<pppd_max>::<NRf>` is the maximum of `pppd`.

## Examples

`READ:PULSe:STATistics:MEASview2:PPPD?` might return -1.1131216288, -151.8063964844,130.5854492188, indicating the pulse-pulse phase difference in dB.

## READ:PULSe:STATistics[:MEASview<y>]:PPPHase (Query Only)

Returns the overshoot measurement result in dB in the pulse statistics measurement. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND` or `TTRend`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:PPPHase
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<pp_phase_avg>, <pp_phase_min>, <pp_phase_max>
```

Where

<pp\_phase\_avg>: : <NRf> is the average pulse-pulse carrier phase.

<pp\_phase\_min>: : <NRf> is the minimum pulse-pulse carrier phase.

<pp\_phase\_max>: : <NRf> is the maximum pulse-pulse carrier phase.

Unit: Degrees.

## Examples

READ:PULSe:STATistics:MEASview2:PPPHase? might return -9.298E-3,-254.3E-3,311.7E-3 for the pulse-pulse carrier phase in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:RINterval (Query Only)

Returns the repetition interval in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:RINterval
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<rep\_int\_avg>, <rep\_int\_min>, <rep\_int\_max>

Where

<rep\_int\_avg>::<NRf> is the average repetition interval.

<rep\_int\_min>::<NRf> is the minimum repetition interval.

<rep\_int\_max>::<NRf> is the maximum repetition interval.

Unit: Seconds.

## Examples

READ:PULSe:STATistics:MEASview2:RINterval? might return 240.5E-6,217.9E-6,281.2E-6 for the repetition interval in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:RIPDb (Query Only)

Returns the ripple in dB in the pulse statistics. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview]:RIPDb

## Related Commands

READ:PULSe:STATistics[:MEASview]:RIPple

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<ripple\_avg>, <ripple\_min>, <ripple\_max>

Where

<ripple\_avg>::<NRf> is the average ripple.

<ripple\_min>::<NRf> is the minimum ripple.

<ripple\_max>::<NRf> is the maximum ripple.

Unit: dB.

## Examples

`READ:PULSe:STATistics:MEASview2:RIPDb?` might return 160.4E-3,83.78E-3,287.7E-3 for the ripple in dB in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:RIPPlE (Query Only)**

Returns the ripple in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview]:RIPPlE
```

## Related Commands

```
READ:PULSe:STATistics[:MEASview]:RIPDb
```

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

```
<ripple_avg>,<ripple_min>,<ripple_max>
```

Where

`<ripple_avg>::<NRf>` is the average ripple.

`<ripple_min>::<NRf>` is the minimum ripple.

`<ripple_max>::<NRf>` is the maximum ripple.

Unit: Percent (%) as watts or volts.

## Examples

`READ:PULSe:STATistics:MEASview2:RIPPlE?` might return 60.4E-3,83.78E-3,287.7E-3 for the ripple in the pulse statistics measurement.

## **READ:PULSe:STATistics[:MEASview<y>]:RISE (Query Only)**

Returns the rise time in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:RISE
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<rise_avg>,<rise_min>,<rise_max>
```

Where

<rise\_avg>::<NRf> is the average rise time.

<rise\_min>::<NRf> is the minimum rise time.

<rise\_max>::<NRf> is the maximum rise time.

Unit: Seconds.

## Examples

READ:PULSe:STATistics:MEASview2:RISE? might return 105.4E-9,91.65E-9, 116.2E-9 for the rise time in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:RMSFreqerror (Query Only)

Returns the RMS frequency error in the pulse statistics measurement. This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TREND.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:RMSFreqerror
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<RMS\_freq\_err\_avg>, <RMS\_freq\_err\_min>, <RMS\_freq\_err\_max>

Where

<RMS\_freq\_err\_avg>::<NRf> is the average of RMS frequency error.

<RMS\_freq\_err\_min>::<NRf> is the minimum of RMS frequency error.

<RMS\_freq\_err\_max>::<NRf> is the maximum of RMS frequency error.

Unit: Hz.

## Examples

READ:PULSe:STATistics:MEASview2:RMSFreqerror? might return 63.67E+3,45.49E+3,81.28E+3 for the RMS frequency error in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:RMSPherror (Query Only)

Returns the RMS phase error in the Pulse Statistics measurement . This command is valid when DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT is set to TRENd.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:RMSPherror

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<RMS\_phase\_err\_avg>, <RMS\_phase\_err\_min>, <RMS\_phase\_err\_max>

Where

<RMS\_phase\_err\_avg>::<NRf> is the average of RMS phase error.

<RMS\_phase\_err\_min>::<NRf> is the minimum of RMS phase error.

<RMS\_phase\_err\_max>::<NRf> is the maximum of RMS phase error.

Unit: Degrees.

## Examples

READ:PULSe:STATistics:MEASview2:RMSPherror? might return 1.032,604.5E-3, 1.606 for the RMS phase error in the pulse statistics measurement.



## READ:PULSe:STATistics[:MEASview<y>]:RRATe (Query Only)

Returns the repetition rate in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

### Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:RRATe
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

```
<rep_rate_avg>,<rep_rate_min>,<rep_rate_max>
```

Where

<rep\_rate\_avg>::<NRf> is the average repetition rate.

<rep\_rate\_min>::<NRf> is the minimum repetition rate.

<rep\_rate\_max>::<NRf> is the maximum repetition rate.

Unit: Hz.

### Examples

`READ:PULSe:STATistics:MEASview2:RRATe?` might return `62.50E+3,62.49E+3,62.52E+3` for the repetition rate in the pulse statistics measurement.

## READ:PULSe:STATistics[:MEASview<y>]:WIDTh (Query Only)

Returns the pulse width in the pulse statistics. This command is valid when `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` is set to `TREND`.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number

### Group

Read commands

### Syntax

```
READ:PULSe:STATistics[:MEASview<y>]:WIDTh
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<width\_avg>, <width\_min>, <width\_max>

Where

<width\_avg>::<NRf> is the average pulse width.

<width\_min>::<NRf> is the minimum pulse width.

<width\_max>::<NRf> is the maximum pulse width.

Unit: Seconds.

## Examples

READ:PULSe:STATistics:MEASview2:WIDTh? might return 4.636E-6,4.630E-6,4.643E-6 for the pulse width in the pulse statistics.

## READ:PULSe:STATistics[:MEASview<y>]:X (Query Only)

Returns the horizontal values of the trace data of the pulse statistics measurement selected by the DISPLAY:PULSe:SElect:RESult command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the DISPLAY:PULSe:STATistics[:MEASview<y>]:PLOT command before executing this query.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

READ:PULSe:STATistics[:MEASview<y>]:X

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

#<num\_digit><num\_byte><data1><data2>....data<n>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

`<data (n) >` is the horizontal value of the statistics graph at the nth data point.

The unit is trace number (Plot = Trend), Hz (Plot = FFT), second (Plot = Time Trend) or current measurement unit (Plot = Histogram).

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:PULSe:STATistics:MEASview2:X?` might return `#264xxxx...` (64-byte data) for the statistics trace of the pulse width measurement when `DISPlay:PULSe:SElect:RESult` is set to `WIDTh`.

## READ:PULSe:STATistics[:MEASview<y>]:XY (Query Only)

Returns the horizontal and vertical values of the trace data of the pulse statistics measurement selected by the `DISPlay:PULSe:SElect:RESult` command.



**Note:** Select the plot type (Trend, FFT, Time Trend or Histogram) using the `DISPlay:PULSe:STATistics[:MEASview<y>]:PLOT` command before executing this query.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number

## Group

Read commands

## Syntax

`READ:PULSe:STATistics[:MEASview<y>]:XY`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`#<num_digit><num_byte><data1><data2>....data<n>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data (n) >` is the horizontal value of the statistics graph at the nth data point.

`<data (n+1) >` is the vertical value of the statistics graph at the nth data point.

For horizontal values, the unit is trace number (Plot = Trend), Hz (Plot = FFT), second (Plot = Time Trend) or current measurement unit (Plot = Histogram).

For vertical values, the unit is current measurement unit (Plot = Trend or Time Trend), dB (Plot = FFT), count (Plot = Histogram)

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:PULSE:STATISTICS:MEASview2:XY?` might return `#264xxxx...` (64-byte data) for the statistics trace of the pulse width measurement when `DISPLAY:PULSE:SELECT:RESULT` is set to `WIDTH`.

## **READ:PULSE:TRACE[:MEASview<y>]:Y**

Returns the vertical values of the pulse trace. Use the `DISPLAY:PULSE:SELECT:NUMBER` command to select the pulse, and the `DISPLAY:PULSE:SELECT:RESULT` command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Read commands

## Syntax

`READ:PULSE:TRACE[:MEASview<y>]:Y?`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

`#<num_digit><num_byte><y(1)><y(2)>...<y(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<y(n)>` is the vertical value of the pulse trace at the `n`th data point, 4-byte little endian floating-point format specified in IEEE 488.2.

The unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The unit can be changed by the `[SENSE]:POWER:UNITS` command.

## Examples

`READ:PULSE:TRACE:MEASVIEW1:Y?` might return `#43204xxxx...` (3204-byte data) for the vertical values of the pulse trace measurement view 1.

## **READ:PULSE:TRACE[:MEASview<y>]:X**

Returns the time values of the pulse trace. Use the `DISPLAY:PULSE:SELECT:NUMBER` command to select the pulse, and the `DISPLAY:PULSE:SELECT:RESULT` command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:TRACe[:MEASview<y>]:X?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
#<num_digit><num_byte><X(1)><X(2)>...<X(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<X(n)> is the time in seconds at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:PULSE:TRACE:MEASVIEW1:X? might return #43204xxxx... (3204-byte data) for the time values of the trace in pulse trace measurement view 1.

## READ:PULSe:TRACe[:MEASview<y>]:XY

Returns the horizontal (time) and vertical value pairs of the pulse trace. Use the DISPLAY:PULSe:SELEct:NUMBER command to select the pulse, and the DISPLAY:PULSe:SELEct:RESUlt command to select the measurement result.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Read commands

## Syntax

```
READ:PULSe:TRACe[:MEASview<y>]:XY?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x (n)><y (n)> is the horizontal value (time in seconds) and vertical value pair at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

The vertical unit depends on measurement results: Hz for frequency error and deviation, degrees for phase error and deviation, otherwise dBm. The vertical unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:PULSE:TRACE:MEASVIEW1:XY? might return #43204xxxx... (3204-byte data) for the horizontal (time) and vertical value pairs of the pulse trace measurement view 1.

## READ:SEM:MHITs:COUNT? (Query Only)

Returns the number of rows in the results table in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Read commands

### Syntax

READ:SEM:MHITs:COUNT?

### Arguments

None

### Returns

<NRf> The number of rows in the table.

## Examples

READ:SEM:MHITs:COUNT? might return 7 indicating there are seven rows in the results table.

## READ:SEM:MHITs<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Read commands

## Syntax

```
READ:SEM:MHITs<x>:AMPLitude:ABSolute?
```

## Arguments

None

## Returns

<NRf> Absolute amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

READ:SEM:MHITs2:AMPLitude:ABSolute? might return -73.17 indicating that the absolute amplitude of signal in row 2 is -73.17 dBm.

## READ:SEM:MHITs<x>:AMPLitude:RELative? (Query Only)

Returns the relative amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

```
READ:SEM:MHITs<x>:AMPLitude:RELative?
```

## Arguments

None

## Returns

<NRf> Relative amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

READ:SEM:MHITs2:AMPLitude:RELative? might return -91.81 indicating that the relative amplitude of mask hit signal in row 2 is -91.81 dBm.

## READ:SEM:MHITs<x>:FREQuency? (Query Only)

Returns the frequency of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

```
READ:SEM:MHITs<x>:FREQuency?
```

## Arguments

None

## Returns

<NRf> Frequency of the specified signal in Hz.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

READ:SEM:MHITs4:FREQuency? might return 2.235E+6 indicating the frequency for mask hit signal in row 4 is 2.235 MHz.

## READ:SEM:MHITs<x>:INTEg:ABSolute? (Query Only)

Returns the absolute integration amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

```
READ:SEM:MHITs<x>:INTEg:ABSolute?
```

## Arguments

None

## Returns

<NRf> Absolute integration amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

READ:SEM:MHITs4:INTEg:ABSolute? might return -75.14 indicating that the absolute integration amplitude for mask hit signal in row 4 is -75.14 dBm.



## **READ:SEM:MHITs<x>:INTEg:RELative? (Query Only)**

Returns the relative integration amplitude of the specified mask hit signal in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Read commands

### **Syntax**

```
READ:SEM:MHITs<x>:INTEg:RELative?
```

### **Arguments**

None

### **Returns**

<NRf> Relative integration amplitude of the specified signal in dBm.

<x> specifies the row in the results table, where the range is from 1 to 12.

### **Examples**

READ:SEM:MHITs4:INTEg:RELative? might return -85.14 indicating that the relative integration amplitude for mask hit signal in row 4 is -85.14 dBm.

## **READ:SEM:MHITs<x>:MARGin:ABSolute? (Query Only)**

Returns the absolute margin for the specified mask hit signal in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Read commands

### **Syntax**

```
READ:SEM:MHITs<x>:MARGin:ABSolute?
```

### **Arguments**

None

### **Returns**

<NRf> Absolute margin of the specified signal in dB.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

`READ:SEM:MHITs4:MARGin:ABSolute?` might return `-62.33` indicating that the absolute margin for mask hit signal in row 4 is `-62.33` dB.

## **READ:SEM:MHITs<x>:MARGin:RELative? (Query Only)**

Returns the relative margin for the specified mask hit signal in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Read commands

### Syntax

`READ:SEM:MHITs<x>:MARGin:RELative?`

### Arguments

None

### Returns

`<NRf>` Relative margin of the specified signal in dB.

`<x>` specifies the row in the results table, where the range is from 1 to 12.

## Examples

`READ:SEM:MHITs4:MARGin:RELative?` might return `-62.33` indicating that the relative margin for mask hit signal in row 4 is `-62.33` dB.

## **READ:SEM:MHITs<x>:OFFSet? (Query Only)**

Returns the offset for the specified mask hit signal in the Spectral Emissions Mask measurement.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Read commands

### Syntax

`READ:SEM:MHITs<x>:OFFSet?`

### Arguments

None

## Returns

<NRf> specifies the mask hit offset.

<x> specifies the row in the results table, where the range is from 1 to 12.

## Examples

READ:SEM:MHITs4:OFFSet? might return BL indicating the offset for mask hit signal in row 4 is BL.

## READ:SEM:PASS? (Query Only)

Returns the pass/fail limit test result in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

READ:SEM:PASS?

## Arguments

None

## Returns

0 (fail) or 1 (pass)

## Examples

READ:SEM:PASS? might return 1, indicating that the limit test was successful.

## READ:SEM:REF:POWER? (Query Only)

Returns the reference power level in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

READ:SEM:REF:POWER?

## Arguments

None

## Returns

<NRf> specifies the reference power.

## Examples

READ:SEM:REF:POWER? might return -16.92 dBm indicating that the reference power level is -16.92 dBm.

## READ:SEM:SPECTrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

READ:SEM:SPECTrum:X?

## Arguments

None

## Returns

#<num\_digit><num\_byte><x(1)><x(2)>...<x(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)> is the frequency (Hz) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:SEM:SPECTRUM:X? might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace.

## READ:SEM:SPECTrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

```
READ:SEM:SPECTrum:XY?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the frequency (Hz) and amplitude (dBm) pair at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:SEM:SPECTRUM:XY? might return #516020xxxx... (16020-byte data) for the frequency and amplitude pairs of the spectrum trace.

## READ:SEM:SPECTrum[:Y]? (Query Only)

Returns the amplitude of the spectrum trace in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Read commands

## Syntax

```
READ:SEM:SPECTrum[:Y]?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><x(2)>...<x(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x (n) > is the frequency (Hz) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:SEM:SPECTRUM[:Y]? might return #516020xxxx... (16020-byte data) for the frequencies of the spectrum trace.

## READ:SGRam[:MEASview<y>] (Query Only)

Returns the spectrogram trace data. The line is selected using the TRACe:SGRam:MEASview<y>:SElect:LINE command.

## Conditions

Measurement views: Spectrogram

## Group

Read commands

## Syntax

READ:SGRam[:MEASview<y>]?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

#<num\_digit><num\_byte><data (1)><data (2)>...<data (n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data (n) > is the trace data in dBm for the point n,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:SGRam:MEASview1? might return #43204xxxx... (3204-byte data) for the spectrogram trace.

## READ:SPECTrum[:MEASview<y>]:TRACe<x>? (Query Only)

Returns the trace data in the Spectrum measurement.



**Note:** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

## Conditions

Measurement views: Spectrum

## Group

Read commands

## Syntax

```
READ:SPECTrum[:MEASview<y>]:TRACe<x>?
```

## Arguments

The parameter <x> = 1 to 5.

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the trace data in dBm for the point n,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:SPECTRUM:MEASview1:TRACE1 might return #43204xxxx... (3204-byte data) for Trace 1 in the Spectrum measurement.

## READ:SPURious:CARRier:POWer? (Query Only)

Returns the carrier power in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

```
READ:SPURious:CARRier:POWer?
```

## Arguments

None

## Returns

<NRf> Carrier power in dBm.

The unit can be changed by the [SENSE]:POWER:UNITS command.

## Examples

READ:SPURIOUS:CARRIER:POWER? might return 4.227, indicating that the carrier power is 4.227 dBm.

## READ:SPURious:COUNT? (Query Only)

Returns the number of spurious signals in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

READ:SPURious:COUNT?

## Arguments

None

## Returns

<NRf> The spurious count.

## Examples

READ:SPURIOUS:COUNT? might return 4, indicating that the spurious count is 4.

## READ:SPURious:PASS? (Query Only)

Returns the pass/fail limit test result in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

READ:SPURious:PASS?



## Arguments

None

## Returns

0 (fail) or 1 (pass).

## Examples

`READ:SPURIOUS:PASS?` might return 1, indicating that the limit test was successful.

## READ:SPURious:SPECTrum:X? (Query Only)

Returns the frequencies of the spectrum trace in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

`READ:SPURious:SPECTrum:X?`

## Arguments

None

## Returns

`#<num_digit><num_byte><x(1)><x(2)>...<x(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<x(n)>` is the frequency (Hz) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:SPURIOUS:SPECTRUM:X?` might return `#516020xxxx...` (16020-byte data) for the frequencies of the spectrum trace in the Spurious measurement.

## READ:SPURious:SPECTrum:XY? (Query Only)

Returns the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

```
READ:SPURious:SPECTrum:XY?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><x(1)><y(1)><x(2)><y(2)>...<x(n)><y(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<x(n)><y(n)> is the frequency (Hz) and amplitude (dBm) pair at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

The amplitude unit can be changed by the [SENSe]:POWER:UNITs command.

## Examples

READ:SPURIOUS:SPECTRUM:XY? might return #516020xxxx... (16020-byte data) for the frequency and amplitude pairs of the spectrum trace in the Spurious measurement.

## READ:SPURious:SPECTrum[:Y]? (Query Only)

Returns the amplitudes of the spectrum trace in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

```
READ:SPURious:SPECTrum[:Y]?
```

## Related Commands

```
READ:SPURious:SPECTrum:X?
```

## Arguments

None

## Returns

#<num\_digit><num\_byte><y(1)><y(2)>...<y(n)>

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<y(n)> is the amplitude (dBm) at the nth data point,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:SPURIOUS:SPECTRUM:Y might return #516020xxxx... (16020-byte data) for the amplitudes of the spectrum trace in the Spurious measurement.

## READ:SPURious:SPUR<x>:AMPLitude:ABSolute? (Query Only)

Returns the absolute amplitude of the specified spurious signal in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

READ:SPURious:SPUR<x>:AMPLitude:ABSolute?

## Arguments

None

## Returns

<NRf> Absolute amplitude of the specified spurious in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:SPURIOUS:SPUR1:AMPLITUDE:ABSOLUTE? might return -19.782, indicating that the absolute amplitude of Spurious #1 is -19.782 dBm.

## **READ:SPURious:SPUR<x>:AMPLitude:RELative? (Query Only)**

Returns the relative amplitude of the specified spurious signal in the Spurious measurement.

### **Conditions**

Measurement views: Spurious

### **Group**

Read commands

### **Syntax**

READ:SPURious:SPUR<x>:AMPLitude:RELative?

### **Arguments**

None

### **Returns**

<NRf> Relative amplitude of the specified spurious signal in dB.

Use the [SENSe]:SPURious:REFeRence command to set the power reference.

### **Examples**

READ:SPURIOUS:SPUR1:AMPLITUDE:RELATIVE? might return -9.782, indicating that the relative amplitude of Spurious #1 is -9.782 dB.

## **READ:SPURious:SPUR<x>:FREQuency:ABSolute? (Query Only)**

Returns the absolute frequency of the specified spurious signal in the Spurious measurement.

### **Conditions**

Measurement views: Spurious

### **Group**

Read commands

### **Syntax**

READ:SPURious:SPUR<x>:FREQuency:ABSolute?

### **Arguments**

None

### **Returns**

<NRf> Absolute frequency of the spurious signal in Hz.

## Examples

`READ:SPURIOUS:SPUR1:FREQUENCY:ABSOLUTE?` might return `2.235E+9`, indicating that the absolute frequency of Spurious #1 is 2.235 GHz.

## **READ:SPURious:SPUR<x>:FREQuency:RELative? (Query Only)**

Returns the relative frequency of the specified spurious signal to the carrier in the Spurious measurement. This command is valid when `[SENSe]:SPURious:REFeRence` is set to `CARRier`.

### Conditions

Measurement views: Spurious

### Group

Read commands

### Syntax

`READ:SPURious:SPUR<x>:FREQuency:RELative?`

### Arguments

None

### Returns

<NRf> Relative frequency of the spurious signal to the carrier in Hz.

## Examples

`READ:SPURIOUS:SPUR1:FREQUENCY:RELATIVE?` might return `3.634E+6`, indicating that the relative frequency of Spurious #1 is 3.634 MHz.

## **READ:SPURious:SPUR<x>:LIMit:ABSolute? (Query Only)**

Returns the absolute amplitude of the limit for the specified spurious signal in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Read commands

### Syntax

`READ:SPURious:SPUR<x>:LIMit:ABSolute?`

### Arguments

None

## Returns

<NRf> Absolute amplitude of the limit for the specified spurious signal in dBm.

The unit can be changed by the [SENSe]:POWer:UNITs command.

## Examples

READ:SPURIOUS:SPUR1:LIMIT:ABSOLUTE? might return -50.0, indicating that the absolute amplitude of the limit for Spurious #1 is -50 dBm.

## READ:SPURious:SPUR<x>:LIMit:RELative? (Query Only)

Returns the relative amplitude of the limit for the specified spurious signal in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

READ:SPURious:SPUR<x>:LIMit:RELative?

## Arguments

None

## Returns

<NRf> Relative amplitude of the limit for the specified spurious signal in dB.

Use the [SENSe]:SPURious:REFerence command to set the power reference.

## Examples

READ:SPURIOUS:SPUR1:LIMIT:RELATIVE? might return -10.0, indicating that the relative amplitude of the limit for Spurious #1 is -10 dB.

## READ:SPURious:SPUR<x>:LIMit:VIOLation? (Query Only)

Returns whether the specified spurious signal exceeds the limit or not.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

```
READ:SPURious:SPUR<x>:LIMit:VIOLation?
```

## Arguments

None

## Returns

0 (under the limit) or 1 (over the limit).

## Examples

READ:SPURIOUS:SPUR1:LIMIT:VIOLATION? might return 1, indicating that Spurious #1 exceeds the limit.

## READ:SPURious:SPUR<x>:RANGe? (Query Only)

Returns the frequency range in which the specified spurious signal occurred.

## Conditions

Measurement views: Spurious

## Group

Read commands

## Syntax

```
READ:SPURious:SPUR<x>:RANGe?
```

## Arguments

None

## Returns

<string> "A" to "T" representing Range A to T, respectively.

## Examples

READ:SPURIOUS:SPUR1:RANGE? might return "E", indicating that Spurious #1 is in Range E.

## READ:SQUality[:MEASview<y>]:FREQUency:DEViation? (Query only)

Returns the frequency deviation in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:FREQuency:DEVIation?
```

## Arguments

None

## Returns

<NRf> Frequency deviation in Hz.

## Examples

READ:SQUality:MEASview1:FREQuency:DEVIation? might return 12.68E+3, indicating the frequency deviation is 12.68 kHz for measurement view 1.

## **READ:SQUality[:MEASview<y>]:FREQuency:DEVIation:TABLE? (Query only)**

Returns the number of columns and the values in the frequency deviation table for a signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:FREQuency:DEVIation:TABLE?
```

## Arguments

None

## Returns

<Dev\_Num>,{<Freq\_dev\_Max>,<Freq\_dev\_Min>,<Freq\_dev\_Avg>}

Where



<Dev\_Num>::<NR1> is the number of columns in the frequency deviation table.

= 2 (2FSK), 4 (4FSK, C4FM), 8 (8FSK), or 16 (16FSK)

<Freq\_dev\_Max>::<NRf> is the maximum frequency deviation in Hz.

<Freq\_dev\_Min>::<NRf> is the minimum frequency deviation in Hz.

<Freq\_dev\_Avg>::<NRf> is the average frequency deviation in Hz.

The dataset <Freq\_dev\_Max>, <Freq\_dev\_Min>, <Freq\_dev\_Avg> is returned for each symbol in ascending order of its level (for example, in order of symbol -3, -1, +1, and +3 for 4FSK).

## Examples

READ:SQUality:MEASview1:FREQuency:DEVIation:TABLE? might return

2,1.257E+3,1.039E+3,1.162E+3,1.586E+3,1.298E+3,1.425E+3 for the frequency signal, populating the results table as follows.

Deviations	-1	+1
Maximum	1.257 kHz	1.586 kHz
Average	1.162 kHz	1.425 kHz
Minimum	1.039 kHz	1.298 kHz

## READ:SQUality[:MEASview<y>]:FREQuency:ERRor? (Query only)

Returns the frequency error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### Group

Read commands

### Syntax

READ:SQUality[:MEASview<y>]:FREQuency:ERRor?

### Arguments

None

### Returns

<NRf> Frequency error in Hz.

## Examples

`READ:SQUALity:MEASview1:FREQUency:ERRor?` might return `612.043E+3`, indicating that the frequency error is 612.043 kHz for measurement view 1.

## **READ:SQUALity[:MEASview<y>]:GAIN:IMBalance? (Query only)**

Returns the gain imbalance in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQUALity[:MEASview<y>]:GAIN:IMBalance?`

## Arguments

None

## Returns

<NRf> Gain imbalance in dB.

## Examples

`READ:SQUALity:MEASview1:GAIN:IMBalance?` might return `-57.746E-3`, indicating that the gain imbalance is -0.057746 dB for measurement view 1.

## **READ:SQUALity[:MEASview<y>]:ORIGin:OFFSet? (Query only)**

Returns the origin offset in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:ORIGin:OFFSet?
```

## Arguments

None

## Returns

<NRf> Origin offset in dB.

## Examples

READ:SQUality:MEASview1:ORIGin:OFFSet? might return -44.968, indicating that the origin offset is -44.968 dB for measurement view 1.

## READ:SQUality[:MEASview<y>]:PEAK:EVM? (Query only)

Returns the peak EVM (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:EVM?
```

## Arguments

None

## Returns

<NRf> Peak EVM in percent (%).

## Examples

READ:SQUality:MEASview1:PEAK:EVM? might return 4.276, indicating that the peak EVM is 4.276% for measurement view 1.

## READ:SQUality[:MEASview<y>]:PEAK:EVM:DB? (Query only)

Returns the peak EVM (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### Group

Read commands

### Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:EVM:DB?
```

### Arguments

None

### Returns

<NRf> Peak EVM in dB.

### Examples

READ:SQUality:MEASview1:PEAK:EVM:DB? might return -27.358, indicating that the peak EVM is -27.358 dB for measurement view 1.

## **READ:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet? (Query only)**

Returns the peak offset EVM (dB) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### Group

Read commands

### Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:EVM:DB:OFFSet?
```

### Arguments

None

## Returns

<NRf> Peak offset EVM in dB.

## Examples

`READ:SQuality:MEASview1:PEAK:EVM:DB:OFFSet?` might return `-37.624`, indicating the peak offset EVM is `-37.624` dB for measurement view 1.

## **READ:SQuality[:MEASview<y>]:PEAK:EVM:LOCation? (Query only)**

Returns the time at which the EVM is peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQuality[:MEASview<y>]:PEAK:EVM:LOCation?`

## Arguments

None

## Returns

<NRf> The time in symbol number at which the EVM is peak.

The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`READ:SQuality:MEASview1:PEAK:EVM:LOCation?` might return `68.000`, indicating that the EVM is peak at symbol #68.000 for measurement view 1.

## **READ:SQuality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet? (Query only)**

Returns the time at which the offset EVM is peak.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:EVM:LOCation:OFFSet?
```

## Arguments

None

## Returns

<NRf> The time in symbol number at which the offset EVM is peak.

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

READ:SQUality:MEASview1:PEAK:EVM:LOCation:OFFSet? might return 123.00, indicating that the offset EVM is peak at symbol #123 for measurement view 1.

## **READ:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet? (Query only)**

Returns the peak offset EVM (%) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:EVM:OFFSet?
```

## Arguments

None

## Returns

<NRf> Peak offset EVM in percent (%).

## Examples

`READ:SQUality:MEASview1:PEAK:EVM:OFFSet?` might return 1.298, indicating the peak offset EVM is 1.298% for measurement view 1.

## **READ:SQUality[:MEASview<y>]:PEAK:FERRor? (Query only)**

Returns the peak FSK error in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to FSK2, FSK4, FSK8, or FSK16.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQUality[:MEASview<y>]:PEAK:FERRor?`

## Arguments

None

## Returns

<NRf> Peak FSK error in percent (%).

## Examples

`READ:SQUality:MEASview1:PEAK:FERRor?` might return 9.136, indicating the peak FSK error is 9.136% for measurement view 1.

## **READ:SQUality[:MEASview<y>]:PEAK:MERRor? (Query only)**

Returns the peak magnitude error (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:MERRor?
```

## Arguments

None

## Returns

<NRf> Peak magnitude error in percent (%).

## Examples

READ:SQUality:MEASview1:PEAK:MERRor? might return 3.595, indicating that the peak magnitude error is 3.595% for measurement view 1.

## **READ:SQUality[:MEASview<y>]:PEAK:MERRor:DB? (Query only)**

Returns the peak magnitude error (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:MERRor:DB?
```

## Arguments

None

## Returns

<NRf> Peak magnitude error in dB.



## Examples

`READ:SQUALity:MEASview1:PEAK:MERRor:DB?` might return `-28.583`, indicating that the magnitude error is `-28.583` dB for measurement view 1.

## **READ:SQUALity[:MEASview<y>]:PEAK:MERRor:LOCation? (Query only)**

Returns the time at which the magnitude error is peak.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQUALity[:MEASview<y>]:PEAK:MERRor:LOCation?`

## Arguments

None

## Returns

`<NRf>` The time in symbol number at which the magnitude error is peak.

The unit can be changed by the `[SENSe]:DDEMod:TIME:UNITs` command.

## Examples

`READ:SQUALity:MEASview1:PEAK:MERRor:LOCation?` might return `68.000`, indicating that the magnitude error is peak at symbol #68 for measurement view 1.

## **READ:SQUALity[:MEASview<y>]:PEAK:PERRor? (Query only)**

Returns the peak phase error in the signal quality measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:PERRor?
```

## Arguments

None

## Returns

<NRf> Peak phase error in degrees.

## Examples

READ:SQUality:MEASview1:PEAK:PERRor? might return 1.907, indicating that the peak phase error is 1.907 ° for measurement view 1.

## READ:SQUality[:MEASview<y>]:PEAK:PERRor:LOCation? (Query only)

Returns the time at which the phase error is peak.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:PEAK:PERRor:LOCation?
```

## Arguments

None

## Returns

<NRf> The time in symbol number at which the phase error is peak.

The unit can be changed by the [SENSe]:DDEMod:TIME:UNITs command.

## Examples

READ:SQUality:MEASview1:PEAK:PERRor:LOCation? might return 68.000, indicating that the phase error is peak at symbol #68 for measurement view 1.

**READ:SQuality[:MEASview<y>]:QUADrature:ERRor? (Query only)**

Returns the quadrature error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

**Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

**Group**

Read commands

**Syntax**

```
READ:SQuality[:MEASview<y>]:QUADrature:ERRor?
```

**Arguments**

None

**Returns**

<NRf> Quadrature error in degrees.

**Examples**

READ:SQuality:MEASview1:QUADrature:ERRor? might return -14.264E-3, indicating that the quadrature error is -0.014264° for measurement view 1.

**READ:SQuality[:MEASview<y>]:RHO? (Query only)**

Returns the  $\rho$  (waveform quality) value in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

**Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

**Group**

Read commands

**Syntax**

```
READ:SQuality[:MEASview<y>]:RHO?
```

## Arguments

None

## Returns

<NRf>  $\rho$  value.

## Examples

READ:SQUality:MEASview1:RHO? might return 998.703E-3, indicating that  $\rho$  is 0.998703 for measurement view 1.

## READ:SQUality[:MEASview<y>]:RMS:EVM? (Query only)

Returns the RMS EVM (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

READ:SQUality[:MEASview<y>]:RMS:EVM?

## Arguments

None

## Returns

<NRf> RMS EVM in percent (%).

## Examples

READ:SQUality:MEASview1:RMS:EVM? might return 2.417, indicating that the RMS EVM is 2.417% for measurement view 1.

## READ:SQUality[:MEASview<y>]:RMS:EVM:DB? (Query only)

Returns the RMS EVM (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQQuality[:MEASview<y>]:RMS:EVM:DB?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB.

## Examples

READ:SQQuality:MEASview1:RMS:EVM:DB? might return -32.356, indicating that the RMS EVM is -32.356 dB for measurement view 1.

## READ:SQQuality[:MEASview<y>]:RMS:EVM:DB:OFFSet? (Query only)

Returns the RMS offset EVM (dB) in the signal quality measurement.

This command is valid when [SENSe]:DDEMod:MODulation:TYPE is set to OQPSK or SOQPSK.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQQuality[:MEASview<y>]:RMS:EVM:DB:OFFSet?
```

## Arguments

None

## Returns

<NRf> RMS offset EVM in dB.

## Examples

`READ:SQQuality:MEASview1:RMS:EVM:DB:OFFSet?` might return `-41.276`, indicating the RMS offset EVM is `-41.276` dB for measurement view 1.

## **READ:SQQuality[:MEASview<y>]:RMS:EVM:OFFSet? (Query only)**

Returns the RMS offset EVM (%) in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to `OQPSK` or `SOQPSK`.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQQuality[:MEASview<y>]:RMS:EVM:OFFSet?`

## Arguments

None

## Returns

`<NRf>` RMS offset EVM in percent (%).

## Examples

`READ:SQQuality:MEASview1:RMS:EVM:OFFSet?` might return `0.783`, indicating the RMS offset EVM is `0.783%` for measurement view 1.

## **READ:SQQuality[:MEASview<y>]:RMS:FERRor? (Query only)**

Returns the RMS FSK error in the signal quality measurement.

This command is valid when `[SENSe]:DDEMod:MODulation:TYPE` is set to `FSK2`, `FSK4`, `FSK8`, or `FSK16`.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:RMS:FERRor?
```

## Arguments

None

## Returns

<NRf> RMS FSK error in percent (%).

## Examples

READ:SQUality:MEASview1:RMS:FERRor? might return 8.67, indicating the RMS FSK error is 8.67% for measurement view 1.

## READ:SQUality[:MEASview<y>]:RMS:MER:DB? (Query only)

Returns the RMS MER (Modulation Error Ratio) in dB in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUality[:MEASview<y>]:RMS:MER:DB?
```

## Arguments

None

## Returns

<NRf> RMS MER in dB.

## Examples

READ:SQUality:MEASview1:RMS:MER:DB? might return 27.394, indicating that the RMS MER is 27.394 dB for measurement view 1.

## **READ:SQQuality[:MEASview<y>]:RMS:MERRor? (Query only)**

Returns the RMS magnitude error (%) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### **Group**

Read commands

### **Syntax**

```
READ:SQQuality[:MEASview<y>]:RMS:MERRor?
```

### **Arguments**

None

### **Returns**

<NRf> RMS magnitude error in percent (%).

### **Examples**

READ:SQQuality:MEASview1:RMS:MERRor? might return 1.837, indicating that the RMS magnitude error is 1.837% for measurement view 1.

## **READ:SQQuality[:MEASview<y>]:RMS:MERRor:DB? (Query only)**

Returns the RMS magnitude error (dB) in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

### **Group**

Read commands

### **Syntax**

```
READ:SQQuality[:MEASview<y>]:RMS:MERRor:DB?
```



## Arguments

None

## Returns

<NRf> RMS magnitude error in dB.

## Examples

`READ:SQUality:MEASview1:RMS:MERRor:DB?` might return `-34.706`, indicating that the magnitude error is `-34.706` dB for measurement view 1.

## **READ:SQUality[:MEASview<y>]:RMS:PERRor? (Query only)**

Returns the RMS phase error in the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58.)

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

`READ:SQUality[:MEASview<y>]:RMS:PERRor?`

## Arguments

None

## Returns

<NRf> RMS phase error in degrees.

## Examples

`READ:SQUality:MEASview1:RMS:PERRor?` might return `893.472E-3`, indicating that the RMS phase error is `0.893472°` for measurement view 1.

## **READ:SQUality[:MEASview<y>]:SYMBOL:LENGth? (Query only)**

Returns the number of analyzed symbols.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Read commands

## Syntax

```
READ:SQUALity[:MEASview<y>]:SYMBOL:LENGTH?
```

## Related Commands

## Returns

<NR1> indicates the length of the synch word in symbols.

## Examples

READ:SQUALity:MEASview1:SYMBOL:LENGTH? might return 3, indicating the length is three symbols for measurement view 1.

## READ:SQUALity[:MEASview<y>]:SYMBOL:RATE? (Query only)

Returns the value of the calculated symbol rate in Hz.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

It is valid when the modulation type is 2|4|8|16FSK and [:SENSe]:DDEMod:SYMBOL:RATE:SEARCh is ON.

## Group

Read commands

## Syntax

```
READ:SQUALity[:MEASview<y>]:SYMBOL:RATE?
```

## Related Commands

[\[:SENSe\]:DDEMod:SYMBOL:RATE:SEARCh](#)

## Returns

<NRf> is the calculated symbol rate in Hz.

## Examples

`READ:SQUality:MEASview1:SYMBOL:RATE?` might return `95.24E+3`, indicating the calculated symbol rate is 95.24 kHz for measurement view 1.

## **READ:SQUality[:MEASview<y>]:SYMBOL:RATE:ERROR? (Query only)**

Returns the value of the symbol rate error in percent (%).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

It is valid when the modulation type is 2|4|8|16FSK and `[ :SENSe ] :DDEMod:SYMBOL:RATE:SEARCh` is ON.

## Group

Read commands

## Syntax

`READ:SQUality[:MEASview<y>]:SYMBOL:RATE:ERROR?`

## Related Commands

[\[SENSe\]:DDEMod:SYMBOL:RATE:SEARCh](#)

## Returns

<NRf> is the symbol error percent (%).

## Examples

`READ:SQUality:MEASview1:SYMBOL:RATE:ERROR?` might return `-0.002`, indicating that the symbol rate error is -0.002% for measurement view 1.

## **READ:TDiagram[:MEASview<y>]:FERRor? (Query only)**

Returns the frequency error in the Trellis diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Read commands

## Syntax

```
READ:TDIagram[:MEASview<y>]:FERRor?
```

## Arguments

None

## Returns

`<freq_error>::<NRf>` is the frequency error in Hz.

## Examples

`READ:TDIagram:MEASview2:FERROR?` might return `-10.7E+3`, indicating the frequency error is `-10.7 kHz`.

## READ:TDIagram[:MEASview<y>]:TRACe? (Query only)

Returns the Trellis diagram trace data.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

## Group

Read commands

## Syntax

```
READ:TDIagram[:MEASview<y>]:TRACe?
```

## Arguments

None

## Returns

```
#<num_digit><num_byte><Y(1)><X(1)><Y(2)><X(2)>...<Y(n)><X(n)>
```

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<Y(n)>` and `<X(n)>` is the phase in degrees and the time in symbols pair at the  $n^{\text{th}}$  data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

`READ:TDIagram:MEASview2:TRACe?` might return `#3160xxxx...` (160-byte data) for the frequency deviation versus Time trace.

## **READ:TOVerview[:MEASview<y>] (Query Only)**

Returns the trace data in the time overview.

### Conditions

Measurement views: Time overview with specified Measurement view number

### Group

Read commands

### Syntax

`READ:TOVerview[:MEASview<y>]`

### Arguments

The parameter `<y>` = 1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

### Returns

`#<num_digit><num_byte><data (1)><data (2)>...<data (n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<data (n)>` is the trace data in dBm for the point n,

4-byte little endian floating-point format specified in IEEE 488.2.

The unit can be changed by the `[SENSe]:POWer:UNITs` command.

### Examples

`READ:TOVerview:MEASview1?` might return `#43204xxxx...` (3204-byte data) for the trace in the time overview.

## **READ:WLAN:AD:CONStE:DATA:EVM? (Query Only)**

Queries the EVM value in Constellation display.

### Conditions

Measurement view: 802.11ad Constellation

### Group

Read commands

## Syntax

READ:WLAN:AD:CONStE:DATA:EVM?

## Arguments

None

## Returns

<NRf> = EVM value

## Examples

READ:WLAN:AD:CONStE:DATA:EVM? might return -42.482, indicating the EVM value present in the analysis data is -42.482 dB.

## READ:WLAN:AD:CONStE:DATA:EVM:PEAK? (Query Only)

Queries the EVM Peak value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Read commands

## Syntax

READ:WLAN:AD:CONStE:DATA:EVM:PEAK?

## Arguments

None

## Returns

<NRf> = EVM peak value

## Examples

READ:WLAN:AD:CONStE:DATA:EVM:PEAK? might return -35.325, indicating the EVM Peak value is -35.325 dB.

## READ:WLAN:AD:CONStE:DATA:EVM:PEAK:SYMBol? (Query Only)

Queries the EVM peak symbol value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Read commands

## Syntax

```
READ:WLAN:AD:CONStE:DATA:EVM:PEAK:SYMBol?
```

## Arguments

None

## Returns

<NRf> = Symbol value

## Examples

READ:WLAN:AD:CONStE:DATA:EVM:PEAK:SYMBOL? might return 6.312, indicating the EVM peak symbol value is 6.312.

## READ:WLAN:AD:CONStE:FERRor? (Query Only)

Queries the frequency Error value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Read commands

## Syntax

```
READ:WLAN:AD:CONStE:FERRor?
```

## Arguments

None

## Returns

<NR3> = frequency error

## Examples

READ:WLAN:AD:CONStE:FERROR? might return 3.344+3, indicating the frequency error value is 3.344 kHz.

## READ:WLAN:AD:CONStE:IQDC:OFFSet? (Query Only)

Queries IQ Origin Offset value in Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Read commands

## Syntax

READ:WLAN:AD:CONStE:IQDC:OFFSet?

## Arguments

None

## Returns

<NRf> = IQ Origin offset value

## Examples

READ:WLAN:AD:CONStE:IQDC:OFFSet? might return -51.841, indicating the IQ Origin Offset value is -51.841 dB.

## READ:WLAN:AD:EVM:TRACe? (Query Only)

Returns the EVM versus time trace data.

## Conditions

Measurement views: 802.11ad EVM vs Time

## Group

Read commands

## Syntax

READ:WLAN:AD:EVM:TRACe?

## Arguments

None

## Returns

#<num\_digit><num\_byte><data(1)><data(2)>...<data(n)>

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data(n)> is the EVM value at the nth data point in percent (%),

<data(n)> follows 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:AD:EVM:TRACe? might return #42036xxxx... (2036-byte data), indicating the EVM versus time trace data.

## READ:WLAN:AD:SUMMArY:AVERAge:BURSt:COUNt (Query Only)

Returns the current burst number out of the total burst in 802.11ad Summary display.



---

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AD:SUMMARY:AVERAGE:BURST:COUNT?
```

## Arguments

None

## Returns

<NR1>

## Examples

READ:WLAN:AD:SUMMARY:AVERAGE:BURST:COUNT might return 1, indicating the current burst count is 1 out of the total burst.

## READ:WLAN:AD:SUMMARY:AVERAGE:BURST:TOTAL:COUNT (Query Only)

Returns the total burst count to be averaged in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AD:SUMMARY:AVERAGE:BURST:TOTAL:COUNT?
```

## Arguments

None

## Returns

<NR1>

## Examples

READ:WLAN:AD:SUMMARY:AVERAGE:BURST:TOTAL:COUNT might return 1, indicating the total burst count to be averaged is 1.

## **READ:WLAN:AD:SUMMARY:ESNR (Query Only)**

Returns the value of estimated SNR value in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Analysis Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:ESNR?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

READ:WLAN:AD:SUMMARY:ESNR might return 36.856, indicating the estimated SNR value is 36.856 dB.

## **READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE (Query Only)**

Returns Avg EVM (Data) value in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE might return 40.7035275551, indicating that the Avg EVM (Data) value is -40.70 dB.

---

## **READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE:STATUS (Query Only)**

Returns Avg EVM (Data) status in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE:STATUS?

### **Arguments**

None

### **Returns**

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

### **Examples**

READ:WLAN:AD:SUMMARY:EVM:DATA:AVERAGE:STATUS might return PASS, indicating that the Avg EVM (Data) test has passed.

## **READ:WLAN:AD:SUMMARY:EVM:DATA:MAX (Query Only)**

Returns Max EVM (Data) value in the 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:EVM:DATA:MAX?

### **Arguments**

None

### **Returns**

<Nrf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:DATA:MAX` might return `922.1968241036E-3`, indicating the Max EVM (Data) value is 922.19 dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:EVM (Query Only)**

Returns the EVM value in the CEF region of the packet in the 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:EVM?`

### Arguments

None

### Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:EVM` might return `6.9680400193`, indicating the EVM value in the CEF region of the packet is 6.69 dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:PEAK (Query Only)**

Returns the peak Error occurred at the given Symbol in CEF Packet in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:PEAK?`

### Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:PEAK` might return 13.7763872743, indicating the peak error value at the given symbol in CEF region of the packet is 13.77 dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:SYMBOL (Query Only)**

Returns the symbol at which the peak error occurred in the CEF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:SYMBOL?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:CEF:SYMBOL` might return 7.402000000E+3, indicating the peak error occurred at symbol 7.40 in the CEF region of the packet.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM (Query Only)**

Returns the EVM value in Data region of the packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM?`

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:EVM might return -33.5668742100, indicating EVM value in Data region of the packet is -33.56 dB.

## READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:PEAK (Query Only)

Returns the peak Error occurred at the given Symbol in Data Packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:PEAK?

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:PEAK might return -26.5964736883, indicating Peak Error value at given Symbol in Data packet is -26.59 dB.

## READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:STATUS (Query Only)

Returns the status of the Data packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKet:DATA:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not Applicable

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:STATUS` might return Pass, indicating that the Data packet test result has Passed.

## **READ:WLAN:AD:SUMMARY:EVM:PACKet:DATA:SYMBOL (Query Only)**

Returns the symbol at which the peak Error occurred in the DATA region of the packet in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKet:DATA:SYMBOL?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:DATA:SYMBOL` might return `4.661000000E+3`, indicating at 4.66 symbol, peak error occurred in Data Region of the packet.

## **READ:WLAN:AD:SUMMARY:EVM:PACKet:GUARd:EVM (Query Only)**

Returns the EVM value in the Guard region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:EVM?

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:EVM might return -65.4983500315, indicating the EVM value in the Guard region of the packet is -65.49 dB.

## READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:PEAK (Query Only)

Returns the peak error occurred at the given symbol in Guard packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:PEAK?

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:PEAK might return -26.5964736883, indicating the peak error value at the given symbol in the Guard packet is -26.59 dB.



---

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:SYMBOL (Query Only)**

Returns the symbol at which the peak error occurred in the Guard region of the packet in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:SYMBOL?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

READ:WLAN:AD:SUMMARY:EVM:PACKET:GUARD:SYMBOL might return 3.334000000E+3, indicating the peak error occurred at symbol 3.33 in the Guard region of the packet.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:EVM (Query Only)**

Returns the EVM value in the Header region of the packet in the 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:EVM?

### **Arguments**

None

### **Returns**

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:EVM` might return `-67.1981081169`, indicating the EVM value in the Header region of the packet is `-67.19` dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:PEAK (Query Only)**

Returns the peak error occurred at the given symbol in the Header packet in the 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:PEAK?`

### Arguments

None

### Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:PEAK` might return `-60.9603086224`, indicating the peak error value at the given symbol in the Header region of the packet is `-60.9603086224`.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:SYMBOL (Query Only)**

Returns the symbol at which the peak error occurred in the Header region of the packet in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:SYMBOL?`

### Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:HEADER:SYMBOL` might return `3.397000000E+3`, indicating the peak error occurred at symbol 3.39 in the Header region of the packet.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:EVM (Query Only)**

Returns the EVM value in STF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:EVM?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:EVM` might return `-22.9729890175`, indicating the EVM value in the STF region of the packet is `-22.97` dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:PEAK (Query Only)**

Returns the peak error at the given symbol in the STF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:PEAK?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:PEAK` might return 38.8123661280, indicating the peak error at the given symbol in the STF packet is 38.81 dB.

## **READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:SYMBOL (Query Only)**

Returns the symbol at which the peak error occurred in the STF region of the packet in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:SYMBOL?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:EVM:PACKET:STF:SYMBOL` might return 0.0000, indicating the peak error occurred at symbol 0.0000 in the STF region of the packet.

## **READ:WLAN:AD:SUMMARY:FERROR:AVERAGE (Query Only)**

Returns the Avg Frequency Error value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:FERRor:AVERage?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:FERROR:AVERAGE` might return `-114.1332879317E-3`, indicating the Avg Frequency Error value is `-114.1332879317E-3` ppm.

## **READ:WLAN:AD:SUMMARY:FERRor:AVERage:STATus (Query Only)**

Returns the Avg Frequency Error status in the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:FERRor:AVERage:STATus?`

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

## Examples

`READ:WLAN:AD:SUMMARY:FERROR:AVERAGE:STATUS` might return `PASS`, indicating that the Avg Frequency Error test has passed.

## **READ:WLAN:AD:SUMMARY:FERRor:HERTZ (Query Only)**

Returns the Frequency Error value in 802.11ad Summary Display in Hz.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:FERRor:HERTZ?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:FERROR:HERTZ` might return `2.875E+3`, indicating the Frequency Error value is `2.875E+3` Hz.

## **READ:WLAN:AD:SUMMARY:FERRor:MAX (Query Only)**

Returns Max Frequency Error value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:FERRor:MAX?`

## Arguments

None

## Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:FERROR:MAX` might return `-114.1332879317E-3`, indicating the maximum frequency error is `-14.1332879317E-3` ppm.

## **READ:WLAN:AD:SUMMARY:FERRor:PPM (Query Only)**

Returns the Frequency Error value in 802.11ad Summary display in ppm.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AD:SUMMARY:FERROR:PPM?
```

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:FERROR:PPM might return 575.023E-3, indicating the Frequency Error value is 575.023E-3 ppm.

## READ:WLAN:AD:SUMMARY:FERROR:STATUS (Query Only)

Returns the Frequency Error Status in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AD:SUMMARY:FERROR:STATUS?
```

## Arguments

None

## Returns

<String>

PASS indicates that the test passed.

FAIL indicates that the test failed.

NA indicates that the test is not applicable.

## Examples

READ:WLAN:AD:SUMMARY:FERROR:STATUS might return PASS, indicating that the Frequency Error test has passed.

## **READ:WLAN:AD:SUMMARY:IQ:GAIN:IMBALANCE (Query Only)**

Returns IQ Gain Imbalance value in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:IQ:GAIN:IMBALANCE?

### **Arguments**

None

### **Returns**

<NRf>

### **Examples**

READ:WLAN:AD:SUMMARY:IQ:GAIN:IMBALANCE might return 69.2700818181E-3, indicating IQ Gain Imbalance value is 69.2700818181E-3 % .

## **READ:WLAN:AD:SUMMARY:IQ:ORIGIN:OFFSET (Query Only)**

Returns IQ Origin Offset value in 802.11ad Summary display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Read commands

### **Syntax**

READ:WLAN:AD:SUMMARY:IQ:ORIGIN:OFFSET?

### **Arguments**

None

### **Returns**

<NRf>



## Examples

`READ:WLAN:AD:SUMMARY:IQ:ORIGIN:OFFSET` might return `-160.0000000000`, indicating that the IQ Origin Offset value is `-160` dB.

## **READ:WLAN:AD:SUMMARY:IQ:QUADRATURE:ERROR (Query Only)**

Returns IQ Quadrature Error value in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:IQ:QUADRATURE:ERROR?`

### Arguments

None

### Returns

<NRf>

## Examples

`READ:WLAN:AD:SUMMARY:IQ:QUADRATURE:ERROR` might return `-220.5967903137E-3`, indicating IQ Quadrature Error value is `-220.5967903137E-3` degrees.

## **READ:WLAN:AD:SUMMARY:PACKET:AGGREGATION (Query Only)**

Returns the value of Aggregation packet information in 802.11ad Summary display.

### Conditions

Measurement view: 802.11ad Summary

### Group

Read commands

### Syntax

`READ:WLAN:AD:SUMMARY:PACKET:AGGREGATION?`

### Arguments

None

## Returns

<Boolean> 0, 1

## Examples

READ:WLAN:AD:SUMMARY:PACKET:AGGREGATION might return "0", indicating the Aggregation value is "0".

## READ:WLAN:AD:SUMMARY:PACKET:BTR (Query Only)

Returns the value of Beam Tracking Request packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:BTR?

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

READ:WLAN:AD:SUMMARY:PACKET:BTR might return "0", indicating the Beam Tracking Request value.

## READ:WLAN:AD:SUMMARY:PACKET:HCS (Query Only)

Returns the value of HCS packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:HCS?

## Arguments

None

## Returns

<String>

## Examples

READ:WLAN:AD:SUMMARY:PACKET:HCS might return "1001010000001000", indicating the HCS value.

## READ:WLAN:AD:SUMMARY:PACKET:LENGTH (Query Only)

Returns the value of Length packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:LENGTH?

## Arguments

None

## Returns

<String>

## Examples

READ:WLAN:AD:SUMMARY:PACKET:LENGTH might return "1000", indicating the Length value is "1000".

## READ:WLAN:AD:SUMMARY:PACKET:MCS (Query Only)

Returns the value of MCS packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:MCS?

## Arguments

None

## Returns

<NR1> Range is 0 to 12

## Examples

READ:WLAN:AD:SUMMARY:PACKET:MCS might return 5, indicating the MCS value is 5.

## READ:WLAN:AD:SUMMARY:PACKET:PPDU (Query Only)

Returns the value of Additional PPDU packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:PPDU?

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

READ:WLAN:AD:SUMMARY:PACKET:PPDU might return "0", indicating the Additional PPDU value is "0".

## READ:WLAN:AD:SUMMARY:PACKET:RSSI (Query Only)

Returns the value of last RSSI packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:RSSI?

## Arguments

None

## Returns

<String>

## Examples

`READ:WLAN:AD:SUMMARY:PACKET:RSSI` might return "0000", indicating the last RSSI value.

## **READ:WLAN:AD:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION (Query Only)**

Returns the value of Scrambler initialization packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION?`

## Arguments

None

## Returns

<String>

## Examples

`READ:WLAN:AD:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION` might return "0101110", indicating the Scrambler Initialization value is "0101110".

## **READ:WLAN:AD:SUMMARY:PACKET:TAROUND (Query Only)**

Returns the value of Turnaround packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

`READ:WLAN:AD:SUMMARY:PACKET:TAROUND?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

READ:WLAN:AD:SUMMARY:PACKET:TAROUND might return "1 ", indicating the Turnaround value.

## READ:WLAN:AD:SUMMARY:PACKET:TLEN (Query Only)

Returns the value of Training Length packet information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:TLEN?

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:PACKET:TLEN might return 21, indicating the value of Training Length packet is 21.

## READ:WLAN:AD:SUMMARY:PACKET:TYPE (Query Only)

Returns the value of Packet Type information in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:PACKET:TYPE?

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

READ:WLAN:AD:SUMMARY:PACKET:TYPE might return "0", indicating the Packet Type value is "0".

## READ:WLAN:AD:SUMMARY:RCPI (Query Only)

Returns the RCPI value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:RCPI?

## Arguments

None

## Returns

<NR1>

## Examples

READ:WLAN:AD:SUMMARY:RCPI might return 207.0000000000, indicating the RCPI value is 207.

## READ:WLAN:AD:SUMMARY:RF:POWER (Query Only)

Queries the RF Output Power value in 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Read commands

## Syntax

READ:WLAN:AD:SUMMARY:RF:POWER?

## Arguments

None

## Returns

<NRf>

## Examples

READ:WLAN:AD:SUMMARY:RF:POWER might return -6.3434391022, indicating the RF Output Power is -6.3434391022 dBm.

## READ:WLAN:AY:CONStE:DATA:EVM? (Query Only)

Query the maximum EVM value present in the analysis data in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Read commands

## Syntax

READ:WLAN:AY:CONStE:DATA:EVM?

## Returns

<NRf>

Units: dB

## Examples

READ:WLAN:AY:CONS:FREQ:ERRO? may return -42.482, which indicates that the maximum EVM value is -42.482 dB.

## READ:WLAN:AY:CONStE:DATA:EVM:PEAK? (Query Only)

Query the Data Peak Error value in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Read commands

## Syntax

READ:WLAN:AY:CONStE:DATA:EVM:PEAK?

## Returns

<NRf>

Units: dB



## Examples

`READ:WLAN:AY:CONS:DATA:EVM:PEAK?` may return -35.325, which indicates that the Data Peak Error value is -35.325 dB.

## **READ:WLAN:AY:CONStE:DATA:EVM:PEAK:SYMBol? (Query Only)**

Query the maximum EVM value at the given Symbol in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Read commands

## Syntax

`READ:WLAN:AY:CONStE:DATA:EVM:PEAK:SYMBol?`

## Returns

<NRf>

## Examples

`READ:WLAN:AY:CONS:DATA:EVM:PEAK:SYMB?` may return 3.327e+003, which is the maximum EVM value at the 3327 Symbol.

## **READ:WLAN:AY:CONStE:FERRor? (Query Only)**

Query the Frequency Error value in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Read commands

## Syntax

`READ:WLAN:AY:CONStE:FERRor?`

## Returns

<NRf>

Units: Hz

## Examples

`READ:WLAN:AY:CONS:FERR?` may return -51.841, which indicates that the Frequency Error value is -51.841 Hz.

## **READ:WLAN:AY:CONStE:FREQuency:ERROr? (Query Only)**

Query the Frequency Error value in the Constellation display.

### **Conditions**

Measurement view: 802.11ay Analysis Constellation

### **Group**

Read commands

### **Syntax**

`READ:WLAN:AY:CONStE:FREQuency:ERROr?`

### **Returns**

<NRf>

Units: Hz

### **Examples**

`READ:WLAN:AY:CONS:FREQ:ERRO?` may return 0.0000, which indicates that the Frequency Error value is 0.0000 Hz.

## **READ:WLAN:AY:CONStE:IQDC:OFFSet? (Query Only)**

Query the IQ Origin Offset value in the Constellation display in dB.

### **Conditions**

Measurement view: 802.11ay Analysis Constellation

### **Group**

Read commands

### **Syntax**

`READ:WLAN:AY:CONStE:IQDC:OFFSet?`

### **Returns**

<NRf>

### **Examples**

`READ:WLAN:AY:CONS:IQDC:OFFS?` may return -51.841, which indicates that the IQ Origin Offset value is -51.841 dB.

## **READ:WLAN:AY:EVM:TRACe? (Query Only)**

Query the EVM versus time trace data after performing a single run.



**Note:** The instrument must be connected.

## Conditions

Measurement views: 802.11ay Analysis EVM

## Group

Read commands

## Syntax

```
READ:WLAN:AY:EVM:TRACe?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the EVM value at the nth data point in percentage (%).

<data(n)> follows 4-byte little endian floating point format specified in IEEE 488.2.

## Examples

READ:WLAN:AY:EVM:TRAC? may return #42036xxxx...(2036-byte data) for the EVM versus time trace.

## READ:WLAN:AY:SUMMary:AVERage:BURSt:COUNT? (Query Only)

Query the current Burst Count in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMary:AVERage:BURSt:COUNT?
```

## Returns

```
<NR1>
```

## Examples

READ:WLAN:AY:SUMM:AVER:BURSt:COUN? may return 1, which indicates that the current Burst Count is 1.

## READ:WLAN:AY:SUMMary:AVERage:BURSt:TOTAL:COUNT? (Query Only)

Query the Total Burst Count to be averaged.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:AVERAGE:BURST:TOTAL:COUNT?`

## Returns

<NR1>

## Examples

`READ:WLAN:AY:SUMM:AVER:BURS:TOTA:COUN?` may return 1, which indicates that the Total Burst Count to be averaged is 1.

## **READ:WLAN:AY:SUMMARY:ESNR? (Query Only)**

Query the Estimated SNR (Signal-to-Noise Ratio) value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:ESNR?`

## Returns

<NRf>

Units: dB

## Examples

`READ:WLAN:AY:SUMM:ESNR?` may return 34.50, which indicates that the Estimated SNR value is 34.50 dB.

## **READ:WLAN:AY:SUMMARY:EVM:DATA:AVERAGE? (Query Only)**

Query the Average EVM (Data) value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMary:EVM:DATA:AVERage?
```

## Returns

<NRf>

Units: dB or % (based on the selected EVM unit)

## Examples

READ:WLAN:AY:SUMM:EVM:DATA:AVER? may return -40.7035275551, which indicates that the Average EVM (Data) value is -40.7035275551 dB.

## READ:WLAN:AY:SUMMary:EVM:DATA:AVERage:STATus? (Query Only)

Query the Average EVM (Data) Status in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMary:EVM:DATA:AVERage:STATus?
```

## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:EVM:DATA:AVER:STAT? may return PASS, FAIL, or NA.

## READ:WLAN:AY:SUMMary:EVM:DATA:MAX? (Query Only)

Query the Maximum EVM (Data) value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:EVM:DATA:MAX?`

## Returns

<NRf>

Units: dB or % (based on the selected EVM unit)

## Examples

`READ:WLAN:AY:SUMMARY:EVM:DATA:MAX?` may return `922.1968241036e-3`, which indicates that the Maximum EVM (Data) value is `922.1968241036e-3` dB.

## **READ:WLAN:AY:SUMMARY:EVM:DATA:SIGMA? (Query Only)**

Query the Standard Deviation (1-Sigma) of the EVM Data.

When the EVM unit is in dB, it queries the Standard Deviation after converting evm values to the dB scale.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:EVM:DATA:SIGMA?`

## Returns

<NRf>

Units: dB

## Examples

`READ:WLAN:AY:SUMMARY:EVM:DATA:SIG?` may return `1.8950`, which indicates that the 1-Sigma EVM Data value is `1.8950` dB.

## **READ:WLAN:AY:SUMMARY:EVM:PACKET:DATA:EVM? (Query Only)**

Query the EVM value that occurs in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMary:EVM:PACKet:DATA:EVM?
```

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

READ:WLAN:AY:SUMM:EVM:PACK:DATA:EVM? may return -33.5668742100, which indicates that the EVM value is -33.5668742100%.

## READ:WLAN:AY:SUMMary:EVM:PACKet:DATA:PEAK? (Query Only)

Query the value of the Peak Error that occurs at the given Symbol in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMary:EVM:PACKet:DATA:PEAK?
```

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

READ:WLAN:AY:SUMM:EVM:PACK:DATA:PEAK? may return -26.5964736883, which indicates that the value of the Peak Error is -26.5964736883 at the given Symbol.

## READ:WLAN:AY:SUMMary:EVM:PACKet:DATA:STATus? (Query Only)

Query the status of the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMMARY:EVM:PACKET:DATA:STATUS?

## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:EVM:PACK:DATA:STAT? may return PASS, FAIL, or NA.

## READ:WLAN:AY:SUMMARY:EVM:PACKET:DATA:SYMBOL? (Query Only)

Query the Symbol at which the Peak Error occurs in the Data Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMMARY:EVM:PACKET:DATA:SYMBOL?

## Returns

<NRf>

## Examples

READ:WLAN:AY:SUMM:EVM:PACK:DATA:SYMB? may return 4.661000000e+3, which indicates that the Peak Error occurs at the 4.661000000e+3 Symbol.

## READ:WLAN:AY:SUMMARY:EVM:PACKET:EDMG:CEF:EVM? (Query Only)

Query the EVM value in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMMARY:EVM:PACKET:EDMG:CEF:EVM?

## Returns

<NRf>



Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:EVM?` may return 6.9680400193, which indicates that the EVM value is -6.9680400193 dB.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:PEAK? (Query Only)**

Query the Peak Error that occurs at the given Symbol in the 11ay 2.16 GHz BW and EDMG CEF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:PEAK?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:PEAK?` may return 13.7763872743, which indicates that the Peak Error value is 13.7763872743% at the given Symbol.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:SYMBOL? (Query Only)**

Query the Symbol at which the Peak Error occurs in the 11ay 2.16 GHz BW and EDMG CEF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:CEF:SYMBOL?`

## Returns

<NRf>

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:CEF:SYMB?` may return `7.402000000e+3`, which indicates that the Peak Error occurs at the `7.402000000e+3` Symbol.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:EDMG:STF:EVM? (Query Only)**

Query the EVM value in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:EDMG:STF:EVM?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:EVM?` may return `-22.9729890175`, which indicates that the EVM value is `-22.9729890175` dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:EDMG:STF:PEAK? (Query Only)**

Query the Peak Error that occurs at the given Symbol in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:EDMG:STF:PEAK?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:PEAK?` may return 38.8123661280, which indicates that the Peak Error value is 38.8123661280% at the given Symbol.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:SYMBOL? (Query Only)**

Query the Symbol at which the Peak Error occurs in the 11ay 2.16 GHz BW and EDMG STF for 11ay 4.32 GHz BW of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:EDMG:STF:SYMBOL?`

## Returns

<Nrf>

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:EDMG:STF:SYMB?` may return 0.0000, which indicates that the Peak Error occurs at the 0.0000 Symbol.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:GUARd:EVM? (Query Only)**

Query the EVM value in the Guard Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:GUARd:EVM?`

## Returns

<Nrf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:GUAR:EVM?` may return `-65.4983500315`, which indicates that the value of the EVM is `-65.4983500315` dB.

## **READ:WLAN:AY:SUMM<sub>ary</sub>:EVM:PACK<sub>et</sub>:GUAR<sub>d</sub>:PEAK? (Query Only)**

Query the value of the Peak Error that occurs at the given Symbol in the Guard Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:GUARd:PEAK?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:GUAR:PEAK?` may return `-26.5964736883`, which indicates that the value of the Peak Error is `-26.5964736883` at the given Symbol.

## **READ:WLAN:AY:SUMM<sub>ary</sub>:EVM:PACK<sub>et</sub>:GUAR<sub>d</sub>:SYMB<sub>ol</sub>? (Query Only)**

Query the Symbol at which the Peak Error occurs in the Guard Region of the packet in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:GUARd:SYMBol?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:GUAR:SYMB?` may return `3.334000000e+3`, which indicates that the Peak Error occurs at the `3.334000000e+3` Symbol.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:HEADer:EVM? (Query Only)**

Query the EVM value in the Header Region of the packet (excluding GI symbols) in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:HEADer:EVM?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:HEAD:EVM?` may return `-67.1981081169`, which indicates that the EVM value in the Header Region is `-67.1981081169` dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:HEADer:PEAK? (Query Only)**

Query the Peak Error that occurs at the given Symbol in Header Region (excluding GI symbols) of the packet in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:HEADer:PEAK?`

## Returns

<NRf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:HEAD:PEAK?` may return `-60.9603086224`, which indicates that the Peak Error value is `-60.9603086224` dB at the given Symbol.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:HEADer:SYMBol? (Query Only)**

Query the Symbol at which the Peak Error occurs in the Header Region of the packet (excluding GI symbols) in the Summary display. This command is not applicable for 11ay 4.32 GHz BW and the value can be ignored.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:HEADer:SYMBol?`

## Returns

<Nrf>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:HEAD:SYMB?` may return `3.3970000000e+3`, which indicates that the peak error occurs at the `3.3970000000e+3` Symbol.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg:A1Header:EVM? (Query Only)**

Query the value of the Pre-EDGM1 Header-A EVMpacket information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg:A1Header:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE1E:A1H:EVM?` may return -34.3702, which indicates that the value of the Pre-EDGM1 Header-A EVM packet is -34.3702 dB.

## **READ:WLAN:AY:SUMM<sub>ary</sub>:EVM:PACK<sub>et</sub>:PRE1Edmg:CEF1:EVM? (Query Only)**

Query the value of the Pre-EDMG1 CEF EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg:CEF1:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE1E:CEF1:EVM?` may return -34.5666, which indicates that the value of the Pre-EDMG1 CEF EVM packet is -34.5666 dB.

## **READ:WLAN:AY:SUMM<sub>ary</sub>:EVM:PACK<sub>et</sub>:PRE1Edmg:GUARd1:EVM? (Query Only)**

Query the value of the Pre-EDMG1 EVM Guard packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE1Edmg:GUARd1:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE1ELGUAR1:EVM?` may return -34.1327, which indicates that the value of the Pre-EDMG1 EVM Guard Region is -34.1327 dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE1Edmg:HEADer1:EVM? (Query Only)**

Query the value of the Pre-EDMG1 L-Header EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE1Edmg:HEADer1:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE1E:HEAD1:EVM?` may return -34.4326, which indicates that the value of the Pre-EDMG1 L-Header EVM packet is -34.4326 dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE1Edmg:STF1:EVM? (Query Only)**

Query the value of the Pre-EDMG1 STF EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE1Edmg:STF1:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)



## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE1E:STF1:EVM?` may return -33.6325, which indicates that the value of the Pre-EDMG1 STF1 EVM packet is -33.6325 dB.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:A2Header:EVM? (Query Only)**

Query the value of the Pre-EDMG2 EDMG Header-A EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:A2Header:EVM?`

### Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE2E:A2H:EVM?` may return -34.37, which indicates that the value of the Pre-EDMG2 EDMG-Header-A EVM packet is -34.37 dB.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:CEF2:EVM? (Query Only)**

Query the value of the Pre-EDMG2 CEF EVM packet information in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:CEF2:EVM?`

### Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE2E:CEF2:EVM?` may return -34.5665, which indicates that the value of the Pre-EDMG2 CEF EVM packet is -34.5665 dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE2Edmg:GUARd2:EVM? (Query Only)**

Query the value of the Pre-EDMG2 Guard EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE2Edmg:GUARd2:EVM?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE2E:GUAR:EVM?` may return -33.137, which indicates that the value of the Pre-EDMG2 Guard EVM packet is -33.137 dB.

## **READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE2Edmg:HEADer2:EVM? (Query Only)**

Query the value of the Pre-EDMG2 L-Header EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:ary:EVM:PACKet:PRE2Edmg:HEADer2:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE2E:HEAD2:EVM?` may return -34.4321, which indicates that the value of the Pre-EDMG2 L-Header EVM packet is -34.4321 dB.

## **READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:STF2:EVM? (Query Only)**

Query the value of the Pre-EDMG2 STF EVM packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:EVM:PACKet:PRE2Edmg:STF2:EVM?`

## Returns

<string>

Units: dB or % (based on the EVM unit selected)

## Examples

`READ:WLAN:AY:SUMM:EVM:PACK:PRE2E:STF2:EVM?` may return -33.6324, which indicates that the value of the Pre-EDMG2 STF2 EVM packet is -33.6324 dB.

## **READ:WLAN:AY:SUMMary:FERRor:AVERage? (Query Only)**

Query the Average Frequency Error in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:FERRor:AVERage?`

## Returns

<NRf>

Units: ppm

## Examples

`READ:WLAN:AY:SUMM:FERR:AVER?` may return `-114.1332879317e-3`, which indicates that the Average Frequency Error value is `-114.1332879317e-3` ppm.

## **READ:WLAN:AY:SUMM:ary:FERRor:AVERage:STATus? (Query Only)**

Query the Average Frequency Error status in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMM:ary:FERRor:AVERage:STATus?`

### Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:FERR:AVER:STAT?` may return PASS, FAIL, or NA.

## **READ:WLAN:AY:SUMM:ary:FERRor:HERTZ? (Query Only)**

Query the Frequency Error value in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMM:ary:FERRor:HERTZ?`

### Returns

<NRf>

Units: Hz or kHz

## Examples

`READ:WLAN:AY:SUMM:FERR:HERT?` may return `2.875e+3`, which indicates that the Frequency Error value is 2.875 kHz.

---

## **READ:WLAN:AY:SUMMary:FERRor:MAX? (Query Only)**

Query the Maximum Frequency Error value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Read commands

### **Syntax**

`READ:WLAN:AY:SUMMary:FERRor:MAX?`

### **Returns**

<NRf>

Units: ppm

### **Examples**

`READ:WLAN:AY:SUMM:FERR:MAX?` may return  $-114.1332879317e-3$ , which indicates that the Maximum Frequency Error value is  $-114.1332879317e-3$  ppm.

## **READ:WLAN:AY:SUMMary:FERRor:PPM? (Query Only)**

Query the Frequency Error value in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Read commands

### **Syntax**

`READ:WLAN:AY:SUMMary:FERRor:PPM?`

### **Returns**

<NRf>

Units: ppm

### **Examples**

`READ:WLAN:AY:SUMM:FERR:PPM?` may return  $575.023e-3$ , which indicates that the Frequency Error value is  $575.023e-3$  ppm.

## **READ:WLAN:AY:SUMMary:FERRor:STATus? (Query Only)**

Query the Frequency Error status in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:FERRor:STATus?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:FERR:STAT?` may return PASS, FAIL, or NA.

## **READ:WLAN:AY:SUMMary:IQ:GAIN:IMBALance? (Query Only)**

Query the IQ Gain Imbalance in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMary:IQ:GAIN:IMBALance?`

## Returns

<NRf>

Units: dB

## Examples

`READ:WLAN:AY:SUMM:IQ:GAIN:IMBAL?` may return `69.2700818181e-3`, which indicates that the IQ Gain Imbalance value is `69.2700818181e-3` dB.

## **READ:WLAN:AY:SUMMary:IQ:ORIGin:OFFSet? (Query Only)**

Query the IQ Origin Offset value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMARY:IQ:ORIGIN:OFFSET?
```

## Returns

<NRf>

Units: dB

## Examples

READ:WLAN:AY:SUMMARY:IQ:ORIGIN:OFFSET? may return -160.0000000000, which indicates that the IQ Origin Offset value is -160.0000000000 dB.

## READ:WLAN:AY:SUMMARY:IQ:QUADRATURE:ERROR? (Query Only)

Query the IQ Quadrature Error value (degrees) in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMARY:IQ:QUADRATURE:ERROR?
```

## Returns

<NRf>

Units: degrees

## Examples

READ:WLAN:AY:SUMMARY:IQ:QUADRATURE:ERROR? may return -220.5967903137e-3, which indicates that the IQ Quadrature Error value is -220.5967903137e-3° degrees.

## READ:WLAN:AY:SUMMARY:PACKET:BEAMFORMED? (Query Only)

Query the value of the Beamformed packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

s

## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:PACK:BEAM? may return 0, which indicates that the value of the Beamformed packet is 0.

## READ:WLAN:AY:SUMM:PACK:BWBitmap? (Query Only)

Query the value of the Bandwidth BitMap packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMM:PACK:BWBitmap?

## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:PACK:BW? may return 11000000, which indicates that the value of the Bandwidth BitMap packet is 11000000.

## READ:WLAN:AY:SUMM:PACK:COMPbw? (Query Only)

Query the value of the Compressed BW packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMM:PACK:COMPbw?



## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:PACK:COMP? may return 2, which indicates that the value of the Compressed BW packet is 2.

## READ:WLAN:AY:SUMM:PACK:CR? (Query Only)

Query the value of the CRC packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMM:PACK:CR?

## Returns

<string>

## Examples

READ:WLAN:AY:SUMM:PACK:CR? may return 0110001001011010, which indicates that the value of the CRC packet is 0110001001011010.

## READ:WLAN:AY:SUMM:PACK:EDMGmcs? (Query Only)

Query the value of the EDGMCS packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

READ:WLAN:AY:SUMM:PACK:EDMGmcs?

## Returns

<NRf>

## Examples

`READ:WLAN:AY:SUMM:PACK:EDMG?` may return 5, which indicates that the value of the EDMG MCS packet is 5.

## **READ:WLAN:AY:SUMM:PACK:LDPCcwlen? (Query Only)**

Query the value of the additional LPDC CW Length packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:PACK:LDPCcwlen?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:PACK:LDPC?` may return 0, which indicates that the value of the LDPC CW Length packet is 0.

## **READ:WLAN:AY:SUMM:PACK:PRIM:channel? (Query Only)**

Query the value of the Primary Channel packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMM:PACK:PRIM:channel?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:PACK:PRIM?` may return 0, which indicates that the value of the Primary Channel packet is 0.

## **READ:WLAN:AY:SUMMary:PACKet:PSDUlen? (Query Only)**

Query the Length value of the packet information in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:AY:SUMMary:PACKet:PSDUlen?
```

### **Returns**

<string>

### **Examples**

READ:WLAN:AY:SUMM:PACK:PSDU? may return 1000, which indicates that the Length value of the packet is 1000.

## **READ:WLAN:AY:SUMMary:PACKet:RSSI? (Query Only)**

Query the value of the last RSSI packet information in the Summary display.

The first 2 bits of the last RSSI will fetch the Guard Interval type for EDMG (00/01/10 - Short/Normal/Long).

### **Conditions**

Measurement view 802.11ay Analysis Summary

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:AY:SUMMary:PACKet:RSSI?
```

### **Returns**

<string>

### **Examples**

READ:WLAN:AY:SUMM:PACK:RSSI? may return 0000, which indicates that the last value of the RSSI packet is 0000.

## **READ:WLAN:AY:SUMMary:PACKet:SCRAmbler:INITialization? (Query Only)**

Query the value of the Scrambler Initialization packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:PACKET:SCRAMBLER:INITIALIZATION?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:PACK:SCRA:INIT?` may return 0101110, which indicates that the value of the Scrambler Initialization packet is 0101110.

## **READ:WLAN:AY:SUMMARY:PACKET:SPATIALSTREAMS? (Query Only)**

Query the value of the Spatial Streams packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

`READ:WLAN:AY:SUMMARY:PACKET:SPATIALSTREAMS?`

## Returns

<string>

## Examples

`READ:WLAN:AY:SUMM:PACK:SPAT?` may return 0, which indicates that the value of the Spatial Streams packet is 0.

## **READ:WLAN:AY:SUMMARY:PACKET:STBCAPPLIED? (Query Only)**

Query the value of the STBC Applied packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMARY:PACKet:STBCapplied?
```

## Returns

```
<string>
```

## Examples

READ:WLAN:AY:SUMM:PACK:STBC? may return 0, which indicates that the value of the STBC Applied packet is 0.

## READ:WLAN:AY:SUMMARY:PACKet:TRNLen? (Query Only)

Query the value of the EDMG TRN Length packet information in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMARY:PACKet:TRNLen?
```

## Returns

```
<string>
```

## Examples

READ:WLAN:AY:SUMM:PAC:TRNL? may return 0, which indicates that the value of the EDMG TRN Length packet is 0.

## READ:WLAN:AY:SUMMARY:RCPI? (Query Only)

Query the RCPI (Received Channel Power Indicator) value in the Summary display.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Read commands

## Syntax

```
READ:WLAN:AY:SUMMARY:RCPI?
```

## Returns

```
<NR1>
```

## Examples

`READ:WLAN:AY:SUMM:RCPI?` may return 207.0000000000, which indicates that the RCPI value is 207.0000000000.

## **READ:WLAN:AY:SUMM:RF:POWer? (Query Only)**

Query the Radio Frequency (RF) Output Power value in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMM:RF:POWer?`

### Returns

<NRf>

You must select the units in the Units tab.

## Examples

`READ:WLAN:AY:SUMM:RF:POW?` may return -6.3434391022, which indicates that the Radio Frequency Output Power value is -6.3434391022.

## **READ:WLAN:AY:SUMM:SIG:QUALity? (Query Only)**

Query the Signal Quality value in the Summary display.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Read commands

### Syntax

`READ:WLAN:AY:SUMM:SIG:QUALity?`

### Returns

<NRf>

Units: rho

## Examples

READ:WLAN:AY:SUMM:SIG:QUA? may return 999.6323e-3, which indicates that the Signal Quality value is 999.6323e-3 rho.

## READ:WLAN:CONStE:MAGNitude? (Query Only)

Returns the magnitude data for the WLAN constellation measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

READ:WLAN:CONStE:MAGNitude?

### Returns

Returned data is in the form

#<num\_digit><num\_byte><data (1)><data (2)>..<data (n)>

where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data (n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

### Examples

READ:WLAN:CONS:MAGN? might return #510400.. (10400-byte data) for the measurement.

## READ:WLAN:CONStE:PHASe? (Query Only)

Returns the phase data for the WLAN Constellation measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

READ:WLAN:CONStE:PHASe?

## Returns

Returned data is in the form

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

where

- <num\_digit> is the number of digits in <num\_byte>
- <num\_byte> is the number of bytes of data that follow
- <data(n)> is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

READ:WLAN:CONS:PHAS? might return #510400.. (10400-byte data) for the measurement.

## READ:WLAN:CONStE:TYPE? (Query Only)

Returns the type (context) of the WLAN Constellation measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:CONStE:TYPE?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers.

The integers must be translated to the proper context as listed below.

Integer value Type (context):

- 0 Pilot
- 1 Data
- 2 Unused or Null

## Examples

READ:WLAN:CONS:TYPE? might return #41352.. (1352-byte data) for the measurement. The actual data must be decoded to the context type using the table above.



## READ:WLAN:CONStE:VALue? (Query Only)

Returns the value (decoded symbol) for the WLAN Constellation measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:CONStE:VALue?
```

### Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in 16-bit little endian integers.

### Examples

READ:WLAN:CONStE:VALue? might return #41352 indicating 1352 bytes of data for the measurement.

## READ:WLAN:CRESPonse:MAGNitude? (Query Only)

Returns the magnitude data for the WLAN Channel Response measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:CRESPonse:MAGNitude?
```

### Returns

Returned data is in the form

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

where

<num\_digit> is the number of digits in <num\_byte>

<num\_byte> is the number of bytes of data that follow

<data (n) > is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

READ:WLAN:CRES:MAGN? might return #510400.. (10400-byte data) for the measurement.

## READ:WLAN:CRESponse:PHASe? (Query Only)

Returns the phase data for the WLAN Channel Response measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

READ:WLAN:CRESponse:PHASe?

### Returns

Returned data is in the form

#<num\_digit><num\_byte><data (1) ><data (2) >..<data (n) >

where

- <num\_digit> is the number of digits in <num\_byte>
- <num\_byte> is the number of bytes of data that follow
- <data (n) > is the data value at the nth data point in dB, 4-byte little endian floating-point format specified in IEEE 488.2

## Examples

READ:WLAN:CRES:PHAS? might return #510400 (10400-byte data) representing the phase data for the measurement.

## READ:WLAN:EVM:TRACe<x>? (Query Only)

Returns the trace data for the selected WLAN EVM trace.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol.

When <x> is 3, the parameter is Average versus Subcarrier

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:EVM:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:EVM:TRAC2? might return #510400.. (10400-byte data) for the measurement.

## READ:WLAN:FLATness:PASS? (Query Only)

Queries whether the average power level values of the subcarriers across the signal bandwidth on the WLAN Spectral Flatness display remain within the limits defined for a particular standard.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Group

Read commands

## Syntax

```
READ:WLAN:FLATness:PASS?
```

## Returns

Returns "1" in the case that all data remains within the defined limits, and "0" otherwise.

## Examples

READ:WLAN:FLAT:PASS? might return 1, which means that the average power level values of subcarriers across the signal bandwidth remains within the defined limits.

## READ:WLAN:FLATness:RESult? (Query Only)

Returns the summary results of each segment of the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:FLATness:RESult?
```

## Returns

The data is returned as 12 comma delineated text values. The specific values are :

- minimum subcarrier within segment 1
- minimum subcarrier average energy deviation from the global average within segment 1
- deviation the of minimum subcarrier average from the allowed value in segment 1
- maximum subcarrier within segment 1
- maximum subcarrier average energy deviation from the global average within segment 1
- deviation of the maximum subcarrier average from the allowed value in segment 1
- minimum subcarrier within segment 2
- minimum subcarrier average energy deviation from the global average within segment 2
- deviation the of minimum subcarrier average from the allowed value in segment 2
- maximum subcarrier within segment 2
- maximum subcarrier average energy deviation from the global average within segment 2
- deviation of the maximum subcarrier average from the allowed value in segment 2

## Examples

READ:WLAN:FLAT:RES? might return -6.0000000000,-17.0626174659E-3,3.9829373825,12.0000000000,19.979260861 which represents the summary results of each segment of the WLAN Spectral Flatness display.

## READ:WLAN:FLATness:TRACe<x>? (Query Only)

Returns the WLAN Spectral Flatness trace data. When <x> is 1, the parameter is Matrix (symbol deviation per subcarrier). When <x> is 2, the parameter is Average Deviation vs Subcarrier.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:FLATness:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:FLAT:TRAC2 might return #510400.. (10400-byte data) for the measurement.

## READ:WLAN:MERRor:TRACe<x>? (Query Only)

Returns the trace data for the selected WLAN Magnitude Error trace.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol.

When <x> is 3, the parameter is Average versus Subcarrier

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:MERRor:TRACe<x>?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:MERR:TRAC2? might return #510400.. (10400-byte data) for the measurement.

## **READ:WLAN:PERRor:TRACe<x>? (Query Only)**

Returns the trace data for the selected WLAN Phase Error trace.

When <x> is 1, the parameter is Matrix.

When <x> is 2, the parameter is Average versus Symbol.

When <x> is 3, the parameter is Average versus Subcarrier

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:PERRor:TRACe<x>?
```

### **Returns**

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point in percent (%), 4-byte little endian floating-point format specified in IEEE 488.2.

### **Examples**

READ:WLAN:PERR:TRAC2? might return #510400.. (10400-byte data) for the measurement.

## **READ:WLAN:PVTime:BURSt:POWer? (Query Only)**

Returns the average power of the burst packet, in dBm, for the WLAN Power vs. Time trace.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:PVTime:BURSt:POWer?
```

### **Returns**

Floating point number that represents the average power of the burst packet, in dBm.

## Examples

`READ:WLAN:PVT:BURST:POWER?` might return `-1.99`, which represents the value of the average power of the burst packet of `-1.99` dBm.

## **READ:WLAN:PVTime:BURSt:WIDTh? (Query Only)**

Returns the value of the measured time width of the burst packet from Power-On to Power-Down (or end of waveform, if that occurs before Power-Down), in seconds, for the WLAN Power vs. Time trace.

## Group

Read commands

## Syntax

`READ:WLAN:PVTime:BURSt:WIDTh?`

## Returns

Floating point number which represents the value of the measured time width of the burst packet from Power-On to Power-Down (or end of waveform, if that occurs before Power-Down), in seconds.

## Examples

`READ:WLAN:PVT:BURST:WIDTH?` might return `156.0000000000E-6`, which represents a burst packet width of `156`  $\mu$ S.

## **READ:WLAN:PVTime:FALL:TIME? (Query Only)**

Queries the value of the width of the falling edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

`READ:WLAN:PVTime:FALL:TIME?`

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

`READ:WLAN:PVT:FALL:TIME?` might return `7e-6`, which indicates the width value of the falling edge is `7`  $\mu$ s.

## **READ:WLAN:PVTime:FALL:TIME:START? (Query Only)**

Queries the value of the start time of the falling edge, in seconds.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:PVTime:FALL:TIME:START?
```

### **Returns**

Floating point number which represents the value of the start time of the falling edge of the burst, in seconds.

### **Examples**

READ:WLAN:PVT:FALL:TIME:STAR? might return 6e-6, which indicates that the start time value of the falling edge is at 6  $\mu$ s.

## **READ:WLAN:PVTime:FALL:TIME:STOP? (Query Only)**

Queries the value of the stop time of the falling edge, in seconds.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:PVTime:FALL:TIME:STOP?
```

### **Returns**

Floating point number which represents the value of the stop time of the falling edge of the burst, in seconds.

### **Examples**

READ:WLAN:PVT:FALL:TIME:STOP? might return 7e-6, which indicates the stop time value of the falling edge is at 7  $\mu$ s.

## **READ:WLAN:PVTime:FTIME? (Query Only)**

Queries the value of the width of the falling edge, in seconds.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:FTIME?
```

## Returns

Floating point number which represents the value of the width of the falling edge of the burst, in seconds.

## Examples

`READ:WLAN:PVT:FALL:TIME?` might return  $7e-6$ , which indicates the width of the falling edge is  $7\ \mu\text{s}$ .

## READ:WLAN:PVTime:RISE:TIME? (Query Only)

Queries the value of the width of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:RISE:TIME?
```

## Returns

Floating point number which represents the value of the width of the rising edge of the burst, in seconds.

## Examples

`FETCH:WLAN:PVT:RISE:TIME?` might return  $7e-6$ , which indicates the width value of the rising edge is  $7\ \mu\text{s}$ .

## READ:WLAN:PVTime:RISE:TIME:START? (Query Only)

Queries the value of the start time of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

`READ:WLAN:PVTime:RISE:TIME:START?`

## Returns

Floating point number which represents the value of the start time of the rising edge of the burst, in seconds.

## Examples

`READ:WLAN:PVT:RISE:TIME:STAR?` might return `6e-6`, which indicates the start time value of the rising edge is at  $6\ \mu\text{s}$ .

## **READ:WLAN:PVTime:RISE:TIME:STOP? (Query Only)**

Queries the value of the stop time of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

`READ:WLAN:PVTime:RISE:TIME:STOP?`

## Returns

Floating point number which represents the value of the stop time of the rising edge of the burst, in seconds.

## Examples

`FETCH:WLAN:PVT:RISE:TIME:STOP?` might return `7e-6`, which indicates the stop time value of the rising edge is at  $7\ \mu\text{s}$ .

## **READ:WLAN:PVTime:RTIME? (Query Only)**

Queries the value of the width of the rising edge, in seconds.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:RTIME?
```

## Returns

Floating point number which represents the value of the width of the rising edge of the burst, in seconds.

## Examples

READ:WLAN:PVT:RISE:TIME? might return 7e-6, which indicates the width value of the rising edge is 7  $\mu$ s.

## READ:WLAN:PVTime:TRACe:X? (Query Only)

Returns the horizontal values (time in seconds) for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:TRACe:X?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>...<data(n)>
```

Where:

<num\_digit> the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:PVT:TRAC:X might return #43204 (3204-byte data), which represents the value of the time in seconds (horizontal values) for the WLAN Power vs. Time trace.

## READ:WLAN:PVTime:TRACe:XY? (Query Only)

Returns the horizontal value (time in seconds) and vertical value (power) for the WLAN Power vs. Time trace.

<x(n)><y(n)> is the horizontal value and vertical value pair at the nth data point.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:TRACe:XY?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

READ:WLAN:PVT:TRAC:XY? might return #43204 (3204-byte data), which represents the horizontal value (time in seconds) and vertical value (power) pair at the nth data point.

## READ:WLAN:PVTime:TRACe[:Y]? (Query Only)

Returns the vertical values (power) for the WLAN Power vs. Time trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:PVTime:TRACe[:Y]?
```

## Returns

```
#<num_digit><num_byte><data(1)><data(2)>..<data(n)>
```

Where:

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

<data(n)> is the data value at the nth data point, 4-byte little endian floating-point format specified in IEEE 488.2.

## Examples

FETCH:WLAN:PVT:TRAC:Y? might return #43204 (3204-byte data), which represents the vertical values (power) for the WLAN Power vs. Time trace.

## READ:WLAN:STABLE:VALUe? (Query Only)

Returns the number of bytes in the symbol table for the WLAN measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:STABLE:VALUe?
```

### Returns

```
#<num_digit><num_byte>
```

Where

<num\_digit> is the number of digits in <num\_byte>.

<num\_byte> is the number of bytes of data that follow.

### Examples

READ:WLAN:STABLE:VALUe? might return #43848 indicating that there are 3848 bytes of data in the symbol table.

## READ:WLAN:SUMMArY:BURSt:INDex? (Query Only)

Returns the index of the analyzed packet within the analysis record.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:SUMMArY:BURSt:INDex?
```

### Arguments

None

### Returns

<NRf> data in dB.

## **READ:WLAN:SUMMARY:BURSt:POWer? (Query Only)**

Returns the average power of all symbols in the packet, including Preamble and Data segments.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:BURSt:POWer?

### **Arguments**

None

### **Returns**

<NRf> data in dBm.

## **READ:WLAN:SUMMARY:BURSt:POWer:CFACTOR? (Query Only)**

Returns the peak-to-average burst power factor in the WLAN measurement. The ratio of the highest instantaneous signal power level to the average signal power.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:BURSt:POWer:CFACTOR?

### **Arguments**

None

### **Returns**

<NRf> data in dB.

## **READ:WLAN:SUMMARY:CERRor? (Query Only)**

Returns the symbol clock error in the WLAN measurement.

Not available for 802.11b.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:SUMMary:CERRor?
```

### Arguments

None

### Returns

<NRf> data in parts per million.

## **READ:WLAN:SUMMary:CPE? (Query Only)**

Returns the RMS magnitude error of the pilots over all data symbols.

Not available for 802.11b.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Read commands

### Syntax

```
READ:WLAN:SUMMary:CPE?
```

### Arguments

None

### Returns

<NRf> data in percent.

## **READ:WLAN:SUMMary:EVM:PEAK:ALL? (Query Only)**

Returns the peak EVM value for all subcarriers in the WLAN measurement or for all chips depending on the selected standard.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:ALL?
```

## Arguments

None

## Returns

<NRf> data in dB.

## **READ:WLAN:SUMMARY:EVM:PEAK:CHIP:ALL? (Query Only)**

Returns the chip at which the reported peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:CHIP:ALL?
```

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:PEAK:CHIP:ONEKchips? (Query Only)**

Returns the chip at which the reported peak EVM value over the first 1000 chips occurred.

Only available for 802.11b.

## Conditions

Measurement view: WLAN



---

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:CHIP:ONEKchips?
```

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:PEAK:DATA? (Query Only)**

Returns the peak EVM value for the data subcarriers in the WLAN measurement.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:DATA?
```

## Arguments

None

## Returns

<NRf> data in dB.

## **READ:WLAN:SUMMARY:EVM:PEAK:ONEKchips? (Query Only)**

Returns the peak EVM value measured over the first 1000 chips.

Available only for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

READ:WLAN:SUMMARY:EVM:PEAK:ONEKchips?

## Arguments

None

## Returns

<NRf> data in dB.

## READ:WLAN:SUMMARY:EVM:PEAK:PILots? (Query Only)

Returns the peak EVM value for the pilots in the WLAN measurement.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

READ:WLAN:SUMMARY:EVM:PEAK:PILots?

## Arguments

None

## Returns

<NRf> data in dB.

## READ:WLAN:SUMMARY:EVM:PEAK:SCARrier:ALL? (Query Only)

Returns the subcarrier on which the ALL peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

---

## Syntax

READ:WLAN:SUMMary:EVM:PEAK:SCARrier:ALL?

## Arguments

None

## Returns

<NR1>

## READ:WLAN:SUMMary:EVM:PEAK:SCARrier:DATA? (Query Only)

Returns the subcarrier on which the DATA peak EVM value occurred.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

READ:WLAN:SUMMary:EVM:PEAK:SCARrier:DATA?

## Arguments

None

## Returns

<NR1>

## READ:WLAN:SUMMary:EVM:PEAK:SCARrier:PILOts? (Query Only)

Returns the subcarrier on which the PILOT peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

READ:WLAN:SUMMary:EVM:PEAK:SCARrier:PILOts?

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:ALL? (Query Only)**

Returns the symbol on which the reported EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:ALL?
```

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:DATA? (Query Only)**

Returns the symbol on which the DATA peak EVM value occurred.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:DATA?
```

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:PILOTS? (Query Only)**

Returns the symbol on which the PILOT peak EVM value occurred.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:PEAK:SYMBOL:PILOTS?
```

## Arguments

None

## Returns

<NR1>

## **READ:WLAN:SUMMARY:EVM:RMS:ALL? (Query Only)**

Returns the RMS EVM data for all subcarriers in the WLAN measurement and for all chips for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:RMS:ALL?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## **READ:WLAN:SUMMARY:EVM:RMS:AVERAGE? (Query Only)**

Returns the average RMS EVM value in the WLAN measurement over the selected number of bursts.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:EVM:RMS:AVERAGE?

### **Arguments**

None

### **Returns**

<NRf> RMS EVM in dB

## **READ:WLAN:SUMMARY:EVM:RMS:DATA? (Query Only)**

Returns the RMS EVM data for the data subcarriers in the WLAN measurement.

Not available for 802.11b.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:EVM:RMS:DATA?

### **Arguments**

None

### **Returns**

<NRf> RMS EVM in dB

## **READ:WLAN:SUMMARY:EVM:RMS:MAXIMUM? (Query Only)**

Returns the maximum RMS EVM value in the WLAN measurement over the selected number of bursts.

---

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:RMS:MAXIMUM?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## **READ:WLAN:SUMMARY:EVM:RMS:ONEKchips? (Query Only)**

Returns the RMS EVM data for the 1k chips in the WLAN measurement measured over the first 1000 chips.

Only available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:RMS:ONEKchips?
```

## Arguments

None

## Returns

<NRf> RMS EVM in dB

## **READ:WLAN:SUMMARY:EVM:RMS:PILots? (Query Only)**

Returns the RMS EVM data for the pilot subcarriers in the WLAN measurement.

Not available for 802.11b.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:EVM:RMS:PILOTS?
```

## Arguments

None

## Returns

<NRf> data in percent.

## READ:WLAN:SUMMARY:FERROR? (Query Only)

Returns the frequency difference between the measured carrier frequency of the signal and the measurement frequency setting.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:FERROR?
```

## Arguments

None

## Returns

<NRf> data in kHz.

## READ:WLAN:SUMMARY:HEADER? (Query Only)

Returns the received packet header information for the WLAN measurement.

Applies to the 802.11b standard only.

## Conditions

Measurement view: WLAN



---

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:HEADER?
```

## Arguments

None

## Returns

Results are returned in the same left-to-right order in which they are presented in the display.

## READ:WLAN:SUMMARY:HTSig? (Query Only)

Returns the decimal values of the received packet HT-SIGNAL. Applies to the 802.11n standard only.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:HTSig?
```

## Arguments

None

## Returns

Results are returned in the same left-to-right order in which they are presented in the display.

## READ:WLAN:SUMMARY:IQOffset? (Query Only)

Returns the average magnitude of the DC subcarrier level relative to total signal power.

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:IQOffset?
```

## Arguments

None

## Returns

<NRf> data in dB.

## **READ:WLAN:SUMMARY:LSIG? (Query Only)**

Returns the decimal values of the received packet (legacy) SIGNAL symbols fields. Applies to the 802.11n standard only.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:LSIG?
```

## Arguments

None

## Returns

Results are returned in the same left-to-right order in which they are presented in the display.

## **READ:WLAN:SUMMARY:PACKET? (Query Only)**

Returns the type and number of symbols, EVM-RMS and average power of the Preamble and Data portions of the packet.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Read commands

## Syntax

```
READ:WLAN:SUMMARY:PACKET?
```

## Arguments

None

**Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

**READ:WLAN:SUMMARY:PACKET:DMODULATION? (Query Only)**

Returns the modulation used in the Data symbols.

**Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

**Group**

Read commands

**Syntax**

```
READ:WLAN:SUMMARY:PACKET:DMODULATION?
```

**Arguments**

None

**Returns**

<string> The data modulation being used.

**READ:WLAN:SUMMARY:PACKET:FORMAT? (Query Only)**

Returns the packet format setting for the WLAN measurement

**Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

**Group**

Read commands

**Syntax**

```
READ:WLAN:SUMMARY:PACKET:FORMAT?
```

**Arguments**

None

**Returns**

<string> The packet format setting.

## **READ:WLAN:SUMMARY:PACKET:GUARD? (Query Only)**

Returns the packet guard interval used by the DATA symbols.

Not available for 802.11b.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:PACKET:GUARD?

### **Arguments**

None

### **Returns**

<NRf> The packet guard interval, 1/4 or 1/8.

## **READ:WLAN:SUMMARY:SIG? (Query Only)**

Returns the decimal values of the received packet SIGNAL symbols fields.

Available only for 802.11agjp.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

READ:WLAN:SUMMARY:SIG?

### **Arguments**

None

### **Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

---

## **READ:WLAN:SUMMARY:VHTSig:A? (Query Only)**

Returns the decimal values of the received packet VHT-SIGNAL(A) symbols' fields.

Available only for 802.11ac.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:SUMMARY:VHTSig:A?
```

### **Arguments**

None

### **Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

## **READ:WLAN:SUMMARY:VHTSig:B? (Query Only)**

Returns the decimal values of the received packet VHT-SIGNAL(B) symbols fields.

Available only for 802.11ac.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Read commands

### **Syntax**

```
READ:WLAN:SUMMARY:VHTSig:B?
```

### **Arguments**

None

### **Returns**

Results are returned in the same left-to-right order in which they are presented in the display.

## Sense commands

### [SENSe]:ACPower:AVERage

Sets or queries how to average waveform in the Channel power and ACPR measurement.

#### Conditions

Measurement views: Channel power and ACPR

#### Group

Sense commands

#### Syntax

```
[SENSe]:ACPower:AVERage { OFF | TIME | FREQuency }  
[SENSe]:ACPower:AVERage?
```

#### Arguments

OFF disables averaging.

TIME averages waveform using time samples.

FREQuency averages waveform using frequency samples.

#### Examples

```
SENSe:ACPOWER:AVERAGE TIME averages waveform using time samples.
```

### [SENSe]:ACPower:AVERage:COUNT

Sets or queries the number of traces for averaging in the Channel power and ACPR measurement.

#### Conditions

Measurement views: Channel power and ACPR

#### Group

Sense commands

#### Syntax

```
[SENSe]:ACPower:AVERage:COUNT <number>  
[SENSe]:ACPower:AVERage:COUNT?
```

#### Arguments

<number>::<NR1> specifies the average count. Range: 2 to 10000.

## Examples

`SENSe:ACPOWER:AVERAGE:COUNT 64` sets the average count to 64.

## [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets `[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO OFF`.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

`[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution] <value>`

`[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]?`

### Related Commands

`[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO`

### Arguments

`<value>::<NRf>` specifies the RBW. Range: 100 Hz to 5 MHz.

## Examples

`SENSe:ACPOWER:BANDWIDTH:RESOLUTION 200kHz` sets the RBW to 200 kHz.

## [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the Channel power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

`[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:ACTual?`

### Arguments

None

## Returns

<NRf> is the actual RBW in Hz.

## Examples

`SENSe:ACPOWER:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

## [SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACPower:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

## Arguments

`OFF` or `0` specifies that the RBW is set manually using the `[SENSe]:ACPower: {BANDwidth|BWIDth}[:RESolution]` command.

`ON` or `1` specifies that the RBW is set automatically.

## Examples

`SENSe:ACPOWER:BANDWIDTH:AUTO ON` sets the RBW automatically.

## [SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets `[SENSe]:ACPower: {BANDwidth|BWIDth}:VIDeo:STATE` `OFF`.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo <value>
```



```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo?
```

## Related Commands

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe
```

## Arguments

<value>::<Nrf> specifies the VBW.

Range: Current RBW/10<sup>4</sup> (1 Hz minimum) to Current RBW.

## Examples

```
SENSe:ACPOWER:BANDWIDTH:VIDEO 200kHz sets the VBW to 200 kHz.
```

## [SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACPower:{BANDwidth|BWIDth}:VIDeo:STATe?
```

## Arguments

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

## Examples

```
SENSe:ACPOWER:BANDWIDTH:VIDEO:STATE ON enables the VBW.
```

## [SENSe]:ACPower:CHANnel:{BANDwidth|BWIDra th}

Sets or queries frequency bandwidth of each channel (all share the same value) in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth} <value>  
[SENSe]:ACPower:CHANnel:{BANDwidth|BWIDth}?
```

## Arguments

<value>::<Nrf> specifies the channel bandwidth. Range: 1 Hz to full span.

## Examples

SENSe:ACPOWER:CHANNEL:BANDWIDTH 1.5MHz sets the channel bandwidth to 1.5 MHz.

## [SENSe]:ACPower:CHANnel:FILTer

Sets or queries the adjacent channel filter in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:CHANnel:FILTer { RRCosine | NONE }  
[SENSe]:ACPower:CHANnel:FILTer?
```

## Arguments

RRCosine uses the Root-Raised Cosine filter.

NONE uses no filter.

## Examples

SENSe:ACPOWER:CHANNEL:FILTER RRCosine uses the Root-Raised Cosine filter for the Channel power and ACPR measurement.

## [SENSe]:ACPower:CHANnel:PAIRs

Sets or queries the number of adjacent channel pairs (upper and lower) in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:CHANnel:PAIRs <number>
```

```
[SENSe]:ACPower:CHANnel:PAIRs?
```

## Arguments

<number>::<NR1> specifies the number of adjacent pairs. Range: 0 to 50.

## Examples

SENSe:ACPOWER:CHANNEL:PAIRS 5 sets five adjacent channel pairs.

## [SENSe]:ACPower:CHANnel:SPACing

Sets or queries frequency difference between centers of each channel in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:CHANnel:SPACing <value>
```

```
[SENSe]:ACPower:CHANnel:SPACing?
```

## Arguments

<value>::<NRf> specifies the channel-to-channel spacing.

Range: 1 Hz to 1 GHz.

## Examples

SENSe:ACPOWER:CHANNEL:SPACING 5MHz sets the channel-to-channel spacing to 5 MHz.

## [SENSe]:ACPower:CHIPrate

Sets or queries the chip rate when [SENSe]:ACPower:CHANnel:FILTer is set to RRCosine (Root Raised Cosine).

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

[SENSe]:ACPower:CHIPrate <value>

[SENSe]:ACPower:CHIPrate?

## Related Commands

[SENSe]:ACPower:CHANnel:FILTer

## Arguments

<value>: : <Nrf> specifies the chip rate. Range: 100 Hz to 105 MHz.

## Examples

SENSe:ACPOWER:CHIPRATE 5kHz sets the chip rate to 5 kHz.

## [SENSe]:ACPower:CLEar:RESults (No Query Form)

Restarts the average trace.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

[SENSe]:ACPower:CLEar:RESults

## Arguments

None

## Examples

SENSe:ACPOWER:CLEAR:RESULTS restarts the average trace.

## [SENSe]:ACPower:FREQuency

Sets or queries the center frequency in the Channel power and ACPR measurement.

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:FREQuency <value>
```

```
[SENSe]:ACPower:FREQuency?
```

## Arguments

<value>::<Nrf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:ACPOWER:FREQUENCY 2.35GHz` sets the center frequency to 2.35 GHz.

## [SENSe]:ACPower:FREQuency:STEP

Sets or queries the frequency step size in the Channel power and ACPR measurement. Programming a specified step size sets

```
[SENSe]:ACPower FREQuency:STEP:AUTO OFF.
```

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACPower:FREQuency:STEP <value>
```

```
[SENSe]:ACPower:FREQuency:STEP?
```

## Related Commands

```
[SENSe]:ACPower:FREQuency:STEP:AUTO
```

## Arguments

<value>::<Nrf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:ACPOWER:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

## [SENSe]:ACPower:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the Channel power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

```
[SENSe]:ACPower:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACPower:FREQuency:STEP:AUTO?
```

### Arguments

OFF or 0 specifies that the frequency step size is set manually using the `[SENSe]:ACPower:FREQuency:STEP` command.

ON or 1 specifies that the frequency step size is set automatically.

## Examples

`SENSe:ACPOWER:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.

## [SENSe]:ACPower:NFLoor:STATe

Determines whether to enable or disable the correction for noise floor.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

```
[SENSe]:ACPower:NFLoor:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACPower:NFLoor:STATe?
```

### Arguments

OFF or 0 disables the correction for noise floor.

ON or 1 enables the correction for noise floor.

## Examples

`SENSe:ACPOWER:NFLLOOR:STATE ON` enables the correction for noise floor.

## [SENSe]:ACPower:POINTs:COUNT

Sets or queries the number of points used for the trace in the Channel Power and ACPR measurement.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

```
[SENSe]:ACPower:POINTs:COUNT { P801 | P2401 | P4001 | P8001 | P10801 }
```

```
[SENSe]:ACPower:POINTs:COUNT?
```

### Arguments

`RTBandwidth` optimizes the measurement for real-time bandwidth.

`DRANge` optimizes the measurement for dynamic range.

## Examples

`SENSe:ACPOWER:POINTS:COUNT P4001` sets the number of points used in the trace of the Channel Power and ACPR display to 4001.

## [SENSe]:ACPower:RRCRolloff

Sets or queries the filter parameter (roll-off ratio) for the Root Raised Cosine filter.

### Conditions

Measurement views: Channel power and ACPR

### Group

Sense commands

### Syntax

```
[SENSe]:ACPower:RRCRolloff <value>
```

```
[SENSe]:ACPower:RRCRolloff?
```

### Related Commands

```
[SENSe]:ACPower:CHANnel:FILTer
```

## Arguments

<value>::<NRf> specifies the filter parameter.

Range: 0.0001 to 1 in 0.0001 steps.

## Examples

`SENSe:ACPOWER:RRCROLLOFF 0.3` sets the filter parameter to 0.3.

## [SENSe]:ACPower:SYMBrate

Sets or queries the symbol rate when [SENSe]:ACPower:CHANnel:FILTer is set to RRCosine (Root Raised Cosine).

## Conditions

Measurement views: Channel power and ACPR

## Group

Sense commands

## Syntax

[SENSe]:ACPower:SYMBrate <value>

[SENSe]:ACPower:SYMBrate?

## Related Commands

[SENSe]:ACPower:CHANnel:FILTer

## Arguments

<value>::<NRf> specifies the symbol rate. Range: 100 Hz to 105 MHz.

## Examples

`SENSe:ACPOWER:CHIPRATE 5kHz` sets the symbol rate to 5 kHz.

## [SENSe]:ACquisition:{BANDwidth|BWIDth}

Sets or queries the acquisition bandwidth (frequency range of the acquisition) when [SENSe]:ACquisition:MODE is set to SAMPlEs or LENGth.

## Conditions

Measurement views: All

## Group

Sense commands



## Syntax

```
[SENSe]:ACQuisition:{BANDwidth|BWIDth} <value>
[SENSe]:ACQuisition:{BANDwidth|BWIDth}?
```

## Arguments

<value>::<NRf> specifies the acquisition bandwidth.

Range: 1MHz to 40 MHz (Standard) / 110MHz (Option 110).

## Examples

SENSe:ACQUISITION:BANDWIDTH 30MHz sets the acquisition bandwidth to 30 MHz.

## [SENSe]:ACQuisition:FFRame:ACTual? (Query Only)

Queries the actual number of Fast Frames.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FFRame:ACTual?
```

## Arguments

None

## Returns

<NR1> The actual number of Fast Frames.

## Examples

SENSe:ACQUISITION:FFRAME:ACTUAL? might return 178, indicating that the actual number of Fast Frames is 178.

## [SENSe]:ACQuisition:FFRame:LIMit

Sets or queries the limit number of Fast Frames.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FFRame:LIMit
```

```
[SENSe]:ACQuisition:FFRame:LIMit?
```

## Arguments

<value>::<NR1> specifies the fast frame limit. Range: 1 to 65535 frames.

Where 1 to 65535 frames for RSA 5000A series. For others, the range is 1 to 220 -1 frames.

## Examples

SENSe:ACQUISITION:FFRAME:LIMIT 500 sets the Fast Frame limit to 500.

## [SENSe]:ACQuisition:FFRame:STATe

Determines whether to enable or disable the Fast Frame.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FFRame:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACQuisition:FFRame:STATe?
```

## Arguments

OFF or 0 disables the Fast Frame.

ON or 1 enables the Fast Frame.

## Examples

SENSe:ACQUISITION:FFRAME:STATE ON enables the Fast Frame.

## [SENSe]:ACQuisition:FSAVe:ENable

Set or queries the FastSave acquisition mode.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FSAVe:ENABle { OFF | ON | 0 | 1 }
[SENSe]:ACQuisition:FSAVe:ENABle?
```

## Arguments

OFF or 0 disables the FastSave acquisition.

ON or 1 enables the FastSave acquisition..

## Examples

SENSe:ACQUISITION:FSAVE:ENABLE ON enables the FastSave acquisition.

## [SENSe]:ACQuisition:FSAVe:FILEs:MAXimum

Set or queries the FastSave maximum number of files saved per run.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FSAVe:FILEs:MAXimum <value>
[SENSe]:ACQuisition:FSAVe:FILEs:MAXimum?
```

## Arguments

<value>::<NR2> specifies the maximum number of files saved per run. Range: 1 to 1,000,000 files.



**Note:** When the FILEs:MAXimum is reached, an event is added to the event queue to indicate that saves are completed. Note that this same event is issued now when Trigger Actions Max total files is reached as well.

## Examples

SENSe:ACQUISITION:FSAVE:FILES:MAXIMUM 10 sets the maximum number of files saved per run to 10.

## [SENSe]:ACQuisition:FSAVe:FORMat

Set or queries the FastSave file format.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FSAVe:FORMat { TIQ| MAT }  
[SENSe]:ACQuisition:FSAVe:FORMat?
```

## Arguments

TIQ selects the TIQ file format for FastSave acquisitions.

MAT selects the MAT file format for FastSave acquisitions.

## Examples

SENSe:ACQUISITION:FSAVE:FORMAT MAT selects the Matlab file format.

## [SENSe]:ACQuisition:FSAVe:LOCation

Set or queries the FastSave file save location (a directory).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FSAVe:LOCation <path>  
[SENSe]:ACQuisition:FSAVe:LOCation?
```

## Arguments

<path>::<string> specifies the location where FastSave acquisitions will be saved.

## Examples

SENSe:ACQUISITION:FSAVE:LOCATION "C:\RSA5100B Files\Test\_folder" sets location where FastSave acquisitions are saved to C:\RSA5100B Files\Test\_folder.

## [SENSe]:ACQuisition:FSAVe:NAME:BASE

Set or queries the FastSave Base file name.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:FSAVe:NAME:BASE <file_name>
```

```
[SENSe]:ACQuisition:FSAVe:NAME:BASE?
```

## Arguments

<file\_name>::<string> specifies the base file name use for FastSave acquisitions.

## Examples

```
SENSe:ACQUISITION:FSAVE:NAME:BASE "FSAVE" sets the Base file name to FSAVE.
```

## [SENSe]:ACQuisition:MEMory:AVAILable:SAMPles? (Query Only)

Returns the amount of acquisition memory available in the instrument.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:MEMory:AVAILable:SAMPles?
```

## Arguments

None

## Returns

<NRf> The amount of acquisition memory available in samples.

## Examples

```
SENSe:ACQUISITION:MEMORY:AVAILABLE:SAMPLES? might return 999.424E+3, indicating that 999424 samples are available.
```

## [SENSe]:ACQuisition:MEMory:CAPacity[:TIME]? (Query Only)

Returns the acquisition memory capacity (maximum period of time that can be acquired with the acquisition memory).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:ACQuisition:MEMory:CAPacity[:TIME]?

## Arguments

None

## Returns

<NRf> The acquisition memory capacity in seconds.

## Examples

SENSe:ACQUISITION:MEMORY:CAPACITY:TIME? might return 26.651E-3, indicating that 26.651 ms can be acquired.

## [SENSe]:ACQuisition:MEMory:USED[:PERCent]? (Query Only)

Returns the percentage of the capacity used based on the current settings.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:ACQuisition:MEMory:USED[:PERCent]?

## Arguments

None

## Returns

<NRf> The percentage of the capacity used.

## Examples

SENSe:ACQUISITION:MEMORY:USED:PERCENT? might return 50.0, indicating that 50% is used.

## [SENSe]:ACQuisition:MODE

Sets or queries the acquisition mode (how to determine the sampling parameters of acquisition bandwidth, samples, and length).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:MODE { AUTO | SAMPlEs | LENGth }
[SENSe]:ACQuisition:MODE?
```

## Related Commands

```
[SENSe]:ACQuisition:{BANDwidth|BWIDth}, [SENSe]:ACQuisition:SAMPlEs,
[SENSe]:ACQuisition:SEConds
```

## Arguments

**AUTO** sets the all sampling parameters automatically.

**SAMPlEs** sets the acquisition bandwidth and samples manually, using the `[SENSe]:ACQuisition:{BANDwidth|BWIDth}` and `:SAMPlEs` commands.

**LENGth** sets the acquisition bandwidth and length manually, using the `[SENSe]:ACQuisition:{BANDwidth|BWIDth}` and `:SEConds` commands.

## Examples

```
SENSe:ACQUISITION:MODE AUTO sets the all sampling parameters automatically.
```

## [SENSe]:ACQuisition:OPTimization

Sets or queries the method of the global gain and input bandwidth optimization.

This command replaces the following commands:

- `[ :SENSe ]:ACPower:OPTimize:SPAN(?)`
- `[ :SENSe ]:DPX:{BANDwidth|BWIDth}:OPTimization(?)`
- `[ :SENSe ]:SGRam:{BANDwidth|BWIDTh}:OPTimization(?)`
- `[ :SENSe ]:SPECTrum:{BANDwidth|BWIDTh}:OPTimization(?)`
- `[ :SENSe ]:MCPower:OPTimize:SPAN(?)`

When a setup or tiq file that was created with an earlier version of software is recalled, the instrument selects a setting for the global optimization that will result in the recalled measurement-specific optimizations. If recalled measurements have conflicting optimizations, then the instrument selects one of the optimizations and issues a GPIB event (event number 108).

## Conditions

Measurement views: Spectrum, DPX, Spectrum, Spectrogram, ACPR, MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:OPTimization { BEST | AUTO | MAXDynrange | MINNoise | MINTime }
```

[SENSe]:ACQuisition:OPTimization?

## Arguments

For RSA7100A only:

**BEST** (Best for multiple displays) causes the instrument to evaluate all the open displays and select an Acquisition Bandwidth wide enough to satisfy as many measurements as possible.

**MINTime** optimizes the gain and input bandwidth to minimize sweep time.

For RSA5000:

**BEST** (Best for multiple displays) causes the instrument to evaluate all the open displays and select an Acquisition Bandwidth wide enough to satisfy as many measurements as possible.

**AUTO** sets the gain and input bandwidth for an optimized trade-off of the competing performance choices.

**MAXDynrange** optimizes the gain and input bandwidth to maximize the dynamic range.

**MINNoise** optimizes the gain and input bandwidth to minimize noise.

**MINTime** optimizes the gain and input bandwidth to minimize sweep time.

## Examples

`SENSE:SPECTRUM:BANDWIDTH:OPTIMIZATION AUTO` optimizes automatically the gain and input bandwidth.

## [SENSe]:ACQuisition:RECORD:ACQuisition:{BANDwidth|BWIDth}:AUTO

Set or queries whether the recording acquisition bandwidth is automatic or manual.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECORD:ACQuisition:{BANDwidth|BWIDth}: AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACQuisition:RECORD:ACQuisition:{BANDwidth|BWIDth}: AUTO ?
```

## Arguments

**ON** or **1** indicates that the acquisition bandwidth during recording is based on the current system acquisition bandwidth.

**OFF** or **0** indicates that the acquisition bandwidth during recording is the selected bandwidth.

## Returns

See Arguments.

## Examples

`SENSe:ACQUISITION:RECORD:ACQUISITION:{BANDWIDTH|BWIDTH}: AUTO` might return `0`, indicating that the acquisition bandwidth during recording is the selected bandwidth.



## [SENSe]:ACQuisition:RECOrd:ACQuisition:{BANDwidth|BWIDth} <value>

Set or queries the Acquisition Bandwidth used during recording.

### Group

Sense commands

### Syntax

```
[SENSe]:ACQuisition:RECOrd:ACQuisition:{BANDwidth|BWIDth} <value> <value>
```

```
[SENSe]:ACQuisition:RECOrd:ACQuisition:{BANDwidth|BWIDth} <value> ?
```

### Arguments

<value>::<NRf> specifies the acquisition bandwidth used during recording.

### Returns

<NRf>

### Examples

SENSe:ACQuisition:RECOrd:ACQuisition:BWIDth 20 sets the acquisition bandwidth to 20 MHz.

## [SENSe]:ACQuisition:RECOrd:FILE:LENGth

Sets or queries how much data (time in milliseconds, seconds, or minutes) is recorded when recording starts. File Length uses the Max saved files per run: setting to determine how many data files are captured per each recording session.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:LENGth <numeric>
```

```
[SENSe]:ACQuisition:RECOrd:FILE:LENGth?
```

### Arguments

<numeric>::=<NR3> determines the file length of each data file saved.

Range: 1E-21 to 10E3 when units set to milliseconds

Range: 1E-5 to 10E3 when units set to seconds

Range: 1E-13 to 10E3 when units set to minutes

## Returns

## Examples

`SENSe:ACQUISITION:RECORD:FILE:LENGTH 10E2` sets the file length to 1000.

## [SENSe]:ACQuisition:RECOrd:FILE:LENGth:ENABle

Sets or queries the file length conditions to determine how much data is recorded. File Length uses theMax saved files per run: setting to determine how many data files are captured per each recording session.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:LENGth:ENABle { OFF | ON | 0 | 1 }
```

```
[SENSe]:ACQuisition:RECOrd:FILE:LENGth:ENABle?
```

## Arguments

ON or 1 enables File Length.

OFF or 0 disables File Length.

## Examples

`SENSe:ACQUISITION:RECORD:FILE:LENGTH:ENABLE ON` enables the File Length conditions.

## [SENSe]:ACQuisition:RECOrd:FILE:LENGth:UNITs

Sets or queries the units used for the file length setting.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:LENGth:UNITs { MSEC | SEC | MIN }
```

## Arguments

MSEC: milliseconds

SEC: seconds

MIN: minutes

## Examples

`SENSe:ACQUISITION:RECORD:FILE:LENGTH:UNITS MSEC` sets the units for the file length to milliseconds.

## [SENSe]:ACQuisition:RECOrd:FILE:LOCation

Sets the file location of the saved data records. To fully support data streaming, the drive selected for recording data must have a minimum write speed of 300 MB/sec.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:LOCation <file_path>
```

```
[SENSe]:ACQuisition:RECOrd:FILE:LOCation?
```

### Arguments

`<file_path>::=<"string">` specifies the location.

If you omit the directory path, the default path is used, which is C:\SignalVu-PC Files\Sample Data Records initially.

### Examples

`SENSe:ACQUISITION:RECORD:FILE:LOCATION C:\SignalVu-PC Files\Sample Data Records` sets the location of the data records to that location.

## [SENSe]:ACQuisition:RECOrd:FILE:MAXimum

Sets or queries how many times a record is created, of the specified File Length setting. If the File Length is not enabled, this setting is ignored. See the command `[ :SENSe]:ACQuisition:RECOrd:FILE:LENGTh:ENABLe`.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:MAXimum <value>
```

[SENSe]:ACQuisition:RECOrd:FILE:MAXimum?

### Arguments

<value>::=<NR1> the number of records created.

Range: 1 to 1000000.

### Examples

SENSe:ACQUISITION:RECORD:FILE:MAXIMUM 10 sets the number of files to saved to ten.

## [SENSe]:ACQuisition:RECOrd:FILE:NAME:BASE

Sets or queries the base file name that is used for all data recordings. The base name is appended with additional information, creating unique file names.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

[SENSe]:ACQuisition:RECOrd:FILE:NAME:BASE <base\_name>

[SENSe]:ACQuisition:RECOrd:FILE:NAME:BASE?

### Arguments

<base\_name>::=<"string"> specifies the base file name used for all data recordings.

### Examples

SENSe:ACQUISITION:RECORD:FILE:NAME:BASE "DataFile" sets the base name to DataFile.

## SENSe:ACQuisition:RECOrd:FILE:NAME:INDEX

Sets or queries the index value of the file suffix.

### Conditions

Requires a RSA7100A my connected and XCOM be the selected file format.

### Group

Sense commands

### Syntax

SENSe:ACQuisition:RECOrd:FILE:NAME:INDEX

## Arguments

{integer value} between 0 and 99999.

## Returns

See arguments

## Examples

`SENSe:ACQUISITION:RECORD:FILE:NAME:INDEX?` might return 12, indicating that the index value is 12.

## SENSe:ACQuisition:RECOrd:FILE:NAME:SUFFix

Sets or queries the file name suffix to append to the output base file name.

## Conditions

Requires a RSA7100A my connected and XCOM be the selected file format.

## Group

Sense commands

## Syntax

`SENSe:ACQuisition:RECOrd:FILE:NAME:SUFFix`

## Arguments

`TIMeStamp` creates a timestamp formed from the file creation time. For example, `myfile-2018.08.15.09.33.12.522.xmrk`.

`NONE` sets the base file name to be used without a suffix. For example, `myfile.xmrk`.

`INDEX` allows you to set an index value. For example, `myfile-00004.xmrk`. Value can be an integer between 0 and 99999.

## Returns

See Arguments.

## Examples

`SENSe:ACQUISITION:RECORD:FILE:NAME:SUFFIX?` might return `INDEX`, indicating that the suffix is set to `Index`.

## [SENSe]:ACQuisition:RECOrd:FILE:SIZE? (Query Only)

Queries the expected recording file size in bytes. This item is only displayed in the UI when Record Time is enabled.

## Group

Sense commands

## Syntax

`[SENSe]:ACQuisition:RECOrd:FILE:SIZE?`

## Arguments

None

## Returns

<NRf> in msec.

## Examples

`SENSe:ACQUISITION:RECORD:FILE:SIZE?` might return 10, indicating that the recorded file size is expected to be 10 msec.

## [SENSe]:ACquisition:RECORD:FILE:STRUCTure

Sets or queries the file structure to use for the recorded data files.

## Conditions

Requires an RSA 306, 306B, 500A series, 600A series, or 7100A be connected.

## Group

Sense commands

## Syntax

```
[SENSe]:ACquisition:RECORD:FILE:STRUCTure { RAW | FORM }
```

## Arguments

RAW: Unformatted files are created. Two files are created for each recording, a header file (.r3h) and the data file (.r3a).

FORMatted: Formatted files are created, ending with a .r3f suffix.

MDSCombined: Midas 2.0 (CDIF) files are created.

MDSDetached: Midas 2.0 Detached (CDIF+DET) files are created.

The contents of these file types is described in the API Reference manual (077-1031-XX) available on the SignalVu-PC Software and Documentation Flash Drive or the Tektronix website [www.Tek.com](http://www.Tek.com).

## Examples

`SENSe:ACQUISITION:RECORD:FILE:STRUCTure FORMATTED` sets the file format to Formatted.

## [SENSe]:ACquisition:RECORD:FILE:WBWidth:DFORmat

Set or queries the data format used for recording. At 800 MHz acquisition bandwidth, data is packed at 12 bits/sample to avoid gaps in the streamed recording. If you require 16-bit samples, use the unpacked argument.

## Conditions

Only available when a RSA7100A is connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECOrd:FILE:WBWidth:DFORmat { PACK | UNPack }
```

```
[SENSe]:ACQuisition:RECOrd:FILE:WBWidth:DFORmat ?
```

## Arguments

PACK : sets data to be recorded in packed data format.

UNPack sets data to be recorded in unpacked data format.

## Returns

See Arguments.

## Examples

SENSe:ACQUISITION:RECORD:FILE:WBWIDTH:DFORMAT might return PACK, indicating that data is set to packed data format.

## [SENSe]:ACQuisition:RECOrd:RESults? (Query Only)

Queries the recording status.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECOrd:RESults?
```

## Arguments

None

## Returns

<Number files completed>,<bytes in file>,<percentage in file>,<Acquisition Data Status>

Where

<Number files completed>::<NR1> Number of files completed.

<bytes in file>::<NR1> Number of bytes written to current file.

<percentage in file>::<NRf> Percentage of current file written.

<Acquisition Data Status>::<NR1> Acquisition data status. This is the overall acquisition data status since recording started.

The following hexadecimal values indicate acquisition data status.

Description	Value
ADC Overrange	0x1
Acquisition Data Lost	0x20
IF Power Overrange	0x100000

### Examples

`SENSe:ACQUISITION:RECORD:RESULTS?` might return `2,2400,50,0x1` indicating that 2 files were completed, 2400 total bytes, 50% completed, and that there is ADC overrange.

## [SENSe]:ACQuisition:RECOrd:STARt (No Query Form)

Starts the data recording. The Record to disk must be set to Record now. See the command `[ :SENSe ]:ACQuisition:RECOrd:TYPE`. See the command `[ :SENSe ]:ACQuisition:RECOrd:FILE:LOCation` to specify the file name and location.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

`[SENSe]:ACQuisition:RECOrd:STARt`

### Arguments

None

### Examples

`SENSe:ACQUISITION:RECORD:START` starts the data recording.

## [SENSe]:ACQuisition:RECOrd:STOP (No Query Form)

Starts the data recording. The Record to disk must be set to Record now. See the command `[ :SENSe ]:ACQuisition:RECOrd:TYPE`. See the command `[ :SENSe ]:ACQuisition:RECOrd:FILE:LOCation` to specify the file name and location.

### Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.

### Group

Sense commands



## Syntax

```
[SENSe]:ACQuisition:RECOrd:STOP
```

## Arguments

None

## Examples

`SENSe:ACQUISITION:RECORD:STOP` stops the data recording.

## [SENSe]:ACQuisition:RECOrd:TYPE

Sets or queries when the recording starts streaming data to the specified file.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:RECOrd:TYPE { NOW | TRIGger }
```

## Arguments

`NOW` starts the recording immediately when Start is enabled. See the command `[ :SENSe]:ACQuisition:RECOrd:START`.

`TRIGger` starts the recording when an appropriate trigger signal is received and Triggering is set to Triggered. See the command `:TRIGger[:SEQuence][:STATus]`.

## Examples

`SENSe:ACQUISITION:RECORD:TYPE NOW` sets the Record to disk setting to Record now.

## [SENSe]:ACQuisition:SAMPles

Sets or queries the acquisition samples (number of samples acquired over the acquisition time) when `[SENSe]:ACQuisition:MODE` is set to `SAMPles`.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ACQuisition:SAMPles <value>
```

[SENSe]:ACQuisition:SAMPles?

### Arguments

<value>::<NR1> specifies the acquisition samples. Range: 2 to 1 G samples..

### Examples

SENSe:ACQUISITION:SAMPLES 1114 sets the acquisition samples to 1114.

## [SENSe]:ACQuisition:SEConds

Sets or queries the acquisition length (time over which the acquisition occurs) when [SENSe]:ACQuisition:MODE is set to LENGth.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

[SENSe]:ACQuisition:SEConds <value>

[SENSe]:ACQuisition:SEConds?

### Arguments

<value>::<NRf> specifies the acquisition length.

### Examples

SENSe:ACQUISITION:SAMPLES 12.5ms sets the acquisition length to 12.5 ms.

## [SENSe]:{AM|FM|PM}:{BANDwidth|BWIDth}:MEASurement

Sets measurement bandwidth for the AM, FM, or PM demodulation to analyze.

### Conditions

Measurement views: AM, FM, PM

### Group

Sense commands

### Syntax

[SENSe]:{AM|FM|PM}:{BANDwidth|BWIDth}:MEASurement <value>

[SENSe]:{AM|FM|PM}:{BANDwidth|BWIDth}:MEASurement?

## Related Commands

FETCH: {AM|FM|PM}?

READ:AM:RESult?

READ:FM:RESult?

READ:PM:RESult?

## Arguments

<value>::<NRF> specifies the demodulation bandwidth for the specified modulation type.

## Examples

SENSe:AM:BANDwidth:MEASurement 1e6 sets the amplitude demodulation bandwidth to 1 MHz.

## [SENSe]:{AM|FM|PM}:CLEar:RESults (No Query Form)

Clear results in the AM/FM/PM measurement.

## Conditions

Measurement views: AM, FM, PM

## Group

Sense commands

## Syntax

[SENSe]: {AM|FM|PM}:CLEar:RESults

## Arguments

None

## Examples

SENSe:AM:CLEAR:RESULTS clears results in the AM measurement.

## [SENSe]:{AM|FM|PM}:{MTPoints|MAXTracepoints}

Sets or queries the maximum trace points in the AM/FM/PM measurement.

## Conditions

Measurement views: AM, FM, PM

## Group

Sense commands

## Syntax

```
[SENSe]:{AM|FM|PM}:{MTPoints|MAXTracepoints} { ONEK | TENK | HUNDredk | NDECimate | NEVerdecimate }
```

```
[SENSe]:{AM|FM|PM}:{MTPoints|MAXTracepoints}?
```

## Arguments

The following table lists the arguments.

Argument	Maximum trace points
ONEK	1k
TENK	10k
HUNDredk	100k
NDECimate or NEVerdecimate	Never decimate

## Examples

`SENSe:AM:MTPOINTS ONEK` selects the maximum trace points of 1000 in the AM measurement.

## [SENSe]:AM:DETECT:AMPLitude

Sets or queries the carrier amplitude detection method used to determine the 0% reference modulation in the AM measurement.

## Conditions

Measurement views: AM

## Group

Sense commands

## Syntax

```
[SENSe]:AM:DETECT:AMPLitude { AVERage | MEDian }
```

```
[SENSe]:AM:DETECT:AMPLitude?
```

## Arguments

`AVERage` defines the 0% reference modulation as the average amplitude in the analysis range (default).

`MEDian` defines the 0% reference modulation as the median amplitude  $((\text{maximum})+(\text{minimum}))/2$  in the analysis range.

## Examples

`SENSe:AM:DETECT:AMPLITUDE AVERage` defines the 0% modulation as the average amplitude in the analysis range.

## [SENSe]:ANALysis:ADVanced:DITHer

Determines whether to enable or disable dithering, or set it automatically.

Dither is a random low-level signal consisting of white noise of one quantizing level peak-to-peak amplitude which may be added to an analog signal prior to sampling for the purpose of minimizing quantization error.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANALysis:ADVanced:DITHer { AUTO | ON | OFF }
```

```
[SENSe]:ANALysis:ADVanced:DITHer?
```

## Arguments

**AUTO** specifies that the dither is set automatically.

**ON** enables dithering.

**OFF** disables dithering.

## Examples

```
SENSe:ANALYSIS:ADVANCED:DITHER ON enables dithering.
```

## [SENSe]:ANALysis:ADVanced:DITHer:HWARe:STATus? (Query Only)

Returns the dithering hardware status.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANALysis:ADVanced:DITHer:HWARe:STATus?
```

## Arguments

None

## Returns

One of the following status information.

**Table 44: Dithering status**

Status	Description
DUNaligned	Dithering is disabled and unaligned.
ON	Dithering is enabled.
OFF	Dithering is disabled.

**Examples**

SENSE:ANALYSIS:ADVANCED:DITHER:HWARE:STATUS? might return OFF, indicating that the dithering is disabled.

**[ :SENSe ]:ANALysis:BANDwidth:FILTer:MODE {set and query}**

Sets or queries the measurement bandwidth filter mode.

**Conditions**

Measurement views: All

**Group**

Sense commands

**Syntax**

SENSe:ANALysis:BANDwidth:FILTer:MODE {0|1|TRUE|FALSE}  
 SENSe:ANALysis:BANDwidth:FILTer:MODE? (Query)

**Returns**

0 is used to change the 'Measurement Bandwidth Filter' checkbox state from unchecked to checked.

1 is used to change 'Measurement Bandwidth Filter' checkbox state from checked to unchecked.

**Examples**

SENSe:ANALysis:BANDwidth:FILTer:MODE 0, changes 'Measurement Bandwidth Filter' checkbox state to unchecked.

**[ SENSe ]:ANALysis:LENGth [ :SOURce<z> ]**

Sets or queries the analysis length. Programming a specified length sets [SENSe]:ANALysis:AUTO OFF.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

**Conditions**

Measurement views: All

**Group**

Sense commands

## Syntax

```
[SENSe]:ANALysis:LENGth[:SOURce2] <value>
```

```
[SENSe]:ANALysis:LENGth[:SOURce2]?
```

## Related Commands

[\[SENSe\]:ANALysis:LENGth\[:SOURce2\]:AUTO](#)

## Arguments

<value> : <NRf> specifies the analysis length.

Range: 10 ns to [(acquisition length) - 400 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

## Examples

SENSe:ANALysis:LENGth:SOURce2 25.625  $\mu$ s sets the analysis length of Source 2 to 25.625  $\mu$ s.

## [SENSe]:ANALysis:LENGth[:SOURce<z>]:ACTual? (Query only)

Queries the actual analysis length.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANALysis:LENGth[:SOURce2]:ACTual?
```

## Returns

<NRf> Actual analysis length in seconds.

## Examples

SENSe:ANALysis:LENGth:SOURce2:ACTual? might return 25.625E-6, indicating that the actual analysis length is 25.625  $\mu$ s.

## [SENSe]:ANALysis:LENGth[:SOURce<z>]:AUTO

Determines whether to set the analysis length automatically or manually.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANALYsis:LENGth:SOURce2:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:ANALYsis:LENGth:SOURce2:AUTO?
```

## Arguments

OFF or 0 sets the analysis length manually, using the [SENSe]:ANALYsis:LENGth command.

ON or 1 sets the analysis length automatically.

## Examples

SENSe:ANALYsis:LENGth:SOURce2:AUTO ON sets the analysis length of Source 2 automatically.

## [SENSe]:ANALYsis:REFeRence[:SOURce<z>]

Sets or queries the analysis time reference.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANALYsis:REFeRence:SOURce2 { ACQSTART | TRIGGER }
```

```
[SENSe]:ANALYsis:REFeRence:SOURce?
```

## Arguments

ACQSTART specifies the acquisition start as the time zero reference.

TRIGGER specifies the trigger point as the time zero reference.

## Examples

SENSe:ANALYsis:REFeRence:SOURce2 ACQSTART specifies the acquisition start as the analysis time reference for Source 2.



## [SENSe]:ANALysis:START[:SOURce<z>]

Sets or queries the analysis offset time. Programming a specified offset time sets [SENSe]:ANALysis:START:AUTO OFF.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:ANALysis:START:SOURCE2 <value>
```

```
[SENSe]:ANALysis:START:SOURCE2?
```

### Related Commands

[\[SENSe\]:ANALysis:LENGth\[:SOURce<z>\]](#)

[\[SENSe\]:ANALysis:START\[:SOURce<z>\]:AUTO](#)

### Arguments

<value>::<Nrf> specifies the analysis offset time.

Range: 0 to [(acquisition length) - 200 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

### Examples

SENSe:ANALysis:START:SOURCE2 23.5µs sets the analysis offset to 23.5 µs.

## [SENSe]:ANALysis:START[:SOURce<z>]:AUTO

Determines whether to set the analysis offset automatically or manually.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

### Conditions

Measurement views: All

### Group

Sense commands

## Syntax

```
[SENSe]:ANALysis:START:SOURCE2:AUTO { OFF | ON | 0 | 1 }  
[SENSe]:ANALysis:SOURCE2:START:AUTO?
```

## Arguments

OFF or 0 sets the analysis offset manually, using the [SENSe]:ANALysis:START command.

ON or 1 sets the analysis offset automatically.

## Examples

SENSe:ANALysis:START:SOURCE1:AUTO ON sets the analysis offset automatically.

## [SENSe]:ANSPectrum:START:AUTO:METHod

Sets or queries the method used for computing auto analysis and spectrum offsets when [SENSe]:ANALysis:REFErence is set to TRIGger.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ANSPectrum:START:AUTO:METHod { INCLude | LEGacy }  
[SENSe]:ANSPectrum:START:AUTO:METHod?
```

## Arguments

INCLude includes the trigger point.

LEGacy starts a trigger point.

## Examples

SENSe:ANSPECTRUM:START:AUTO:METHOD INCLUDE includes the trigger point.

## [SENSe]:AUDio:AVERage:COUNT

Sets or queries the number of harmonics to average in the audio measurement.

The range is from 2 to 20.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:AVERAge:COUNT <average_value>
```

```
[SENSe]:AUDio:AVERAge:COUNT?
```

## Arguments

<average\_value> ::= <NR1> the number of harmonics to average.

## Examples

SENSe:AUDIO:AVERAGE:COUNT 10 sets the number of harmonics to average to 10.

## [SENSe]:AUDio:AVERAge:ENABLE

Enables or disables averaging of harmonics in the audio measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:AVERAge:ENABLE { OFF | ON | 0 | 1 }
```

```
[SENSe]:AUDio:AVERAge:ENABLE?
```

## Arguments

OFF or 0 turns averaging off.

ON or 1 turns averaging on.

## Examples

SENSe:AUDIO:AVERAGE:ENABLE ON turns on averaging of harmonics in the audio measurement.

## [SENSe]:AUDio:BANDwidth|BWIDth

Sets or queries the audio measurement bandwidth.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:BANDwidth|BWIDth <value>
```

```
[SENSe]:AUDio:BANDwidth|BWIDth?
```

## Arguments

<value> ::= <NR3> the audio measurement bandwidth value.

## Examples

SENSe:AUDIO:BANDWIDTH|BWIDTh 20.5E+3 sets the audio measurement bandwidth to 20.5 kHz.

## [SENSe]:AUDio:BANDwidth|BWIDth:RESolution

Sets or queries the resolution bandwidth for the audio measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:BANDwidth|BWIDth:RESolution <value>
```

```
[SENSe]:AUDio:BANDwidth|BWIDth:RESolution?
```

## Arguments

<value> ::= <NR3> the resolution bandwidth value.

## Examples

SENSe:AUDIO:BANDWIDTH|BWIDTh:RESOLUTION 40 sets the resolution bandwidth to 40 Hz.

## [SENSe]:AUDio:BANDwidth|BWIDth:RESolution:AUTO

Determines whether to set the resolution bandwidth frequency automatically or manually.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:BAWdwidth|BWIDth:RESolution:AUTO { OFF | ON | 0 | 1 }
[SENSe]:AUDio:BAWdwidth|BWIDth:RESolution:AUTO?
```

## Arguments

OFF or 0 specifies the resolution bandwidth is set manually.

ON or 1 specifies the resolution bandwidth is set automatically.

## Examples

SENSe:AUDIO:BAWdwidth|BWIDth:RESolution:AUTO ON sets the resolution bandwidth automatically.

## [SENSe]:AUDio:BAWdwidth|BWIDth:RESolution:SHAPE

Sets or queries the shape of the resolution bandwidth for the audio measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:BAWdwidth|BWIDth:RESolution:SHAPE { FLATtop | KAISer }
[SENSe]:AUDio:BAWdwidth|BWIDth:RESolution:SHAPE?
```

## Arguments

FLATtop sets the resolution bandwidth shape to Flattop.

KAISer sets the resolution bandwidth shape to Kaiser.

## Examples

SENSe:AUDIO:BAWdwidth|BWIDth:RESolution:SHAPE FLATtop sets the resolution bandwidth to Flattop for the audio measurement.

## [SENSe]:AUDio:CARRier:OFFSet

Sets or queries the carrier frequency offset. This selection is available for FM or PM measurements.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:CARRier:OFFSet <value>
```

```
[SENSe]:AUDio:CARRier:OFFSet?
```

## Arguments

<value> ::= <NR3> the carrier frequency offset value.

## Returns

<NR3>

## Examples

```
SENSe:AUDIO:CARRIER:OFFSET 422.5 sets the carrier frequency offset to 422.5 Hz.
```

## [SENSe]:AUDio:CARRier:OFFSet:AUTO

Determines whether to set the carrier frequency error automatically or manually. This selection is available FM or PM measurements.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:CARRier:OFFSet:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:AUDio:CARRier:OFFSet:AUTO?
```

## Arguments

OFF or 0 specifies the carrier frequency error is set manually.

ON or 1 specifies the carrier frequency error is set automatically.

## Examples

```
SENSe:AUDIO:CARRIER:OFFSET:AUTO ON sets the carrier frequency error automatically.
```

## [SENSe]:AUDio:DEMod:MUTE

Sets or queries the mute setting for the audio demodulation. Enabling Mute inhibits the audio from playing out through the PC speakers.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:DEMod:MUTE { OFF | ON | 0 | 1 }
```

## Arguments

ON or 1 enables the Mute setting for audio demodulation.

OFF or 0 disables the Mute setting for audio demodulation.

## Returns

ON or 1 means that the Mute setting for audio demodulation is enabled.

OFF or 0 means that the Mute setting for audio demodulation is disabled.

## Examples

SENSe:AUDIO:DEMOD:MUTE 1 enables the audio demodulation mute setting, inhibiting the audio signal from playing out through the PC speakers.

## [SENSe]:AUDio:DEMod[:STATe]

Sets or queries the audio demodulation state.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:DEMod[:STATe] { 0 | 1 }
```

## Arguments

1 starts the demodulation.

0 stops the demodulation.

## Returns

1 means the demodulation is started.

0 means the demodulation is stopped.

## Examples

SENSe:AUDIO:DEMOD[:STATE] 1 starts the demodulating the audio signal.

## **[SENSe]:AUDio:DEMod:STReam:ACTivate**

Sets or queries the Stream to File function. When checked, anytime the Run button is pressed, the audio stream-to-file begins.

### **Conditions**

Requires an RSA be connected with SignalVu-PC.

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:AUDio:DEMod:STReam:ACTivate { OFF | ON | 0 | 1 }  
[SENSe]:AUDio:DEMod:STReam:ACTivate?
```

### **Arguments**

ON or 1 enables the stream to file function.

OFF or 0 disables the stream to file function.

### **Returns**

ON or 1 means the stream to file function is enabled.

OFF or 0 means the stream to file function is disabled.

### **Examples**

`SENSe:AUDIO:DEMOD:STREAM:ACTIVATE 1` enables (activates) the audio streaming function.

`SENSe:AUDIO:DEMOD:STREAM:ACTIVATE?` might return 0, indicating the audio streaming function is deactivated.

## **[SENSe]:AUDio:DEMod:STReam[:FILE] (No Query Form)**

Sets the file name and file location of the audio file when streaming to disc.

### **Conditions**

Requires an RSA be connected with SignalVu-PC.

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:AUDio:DEMod:STReam[:FILE] <file>
```

### **Arguments**

`<file_name>::=<string>` specifies the file name and location. The file extension is `.wav`. You can omit the extension.



You can use the absolute path to specify the file name. For example, specify the SAMPLE1 file in the My Documents folder on the C drive as "C:\My Documents\Audio1".

If you omit the directory path, the default path is used, which is C:\SignalVu-PC Files\Sample Data Records, initially.

## Examples

`SENSe:AUDIO:DEMOD:STREAM[:FILE]` "C:\SignalVu-PC Files\Sample Data Records\Audio1" saves the streamed audio file Audio1 to the specified location.

## [SENSe]:AUDio:DEMod:TYPE

Sets or queries both the audio signal demodulation type and bandwidth.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:DEMod:TYPE { AM8K | FM8K | FM13K | FM75K | FM200K }
```

## Related Commands

### Arguments

AM8K

FM8K

FM13K

FM75K

FM200K

### Returns

Same as arguments.

### Examples

`SENSe:AUDIO:DEMOD:TYPE AM8K` sets the audio demodulation type to AM and the bandwidth to 8 KHz.

`SENSe:AUDIO:DEMOD:TYPE?` might return AM8K.

## [SENSe]:AUDio:DEMod:VOLume

Sets or queries the volume of the demodulated audio signal when played out through the PC speakers.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Sense commands

## Syntax

```
[SENSE]:AUDIO:DEMOD:VOLUME <value>
```

```
[SENSe]:AUDIO:DEMOD:VOLUME?
```

## Arguments

<value> ::= <NR2> the value of relative volume setting.

Range = 0.0 to 1.0.

## Examples

SENSe:AUDIO:DEMOD:VOLUME 1.0 sets the volume output to maximum.

## [SENSe]:AUDIO:FILTer:DEEMphasis

Sets or queries the de-emphasis filter type.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSE]:AUDIO:FILTer:DEEMphasis { U25 | U50 | U75 | U750 | USER }
```

```
[SENSe]:AUDIO:FILTer:DEEMphasis?
```

## Arguments

U25 specifies the 25  $\mu$ s filter.

U50 specifies the 50  $\mu$ s filter.

U75 specifies the 75  $\mu$ s filter.

U750 specifies the 750  $\mu$ s filter.

USER specifies a user-defined filter.

## Examples

SENSe:AUDIO:FILTer:DEEMPHASIS U50 specifies the 50  $\mu$ s filter for the audio measurement.

## [SENSe]:AUDio:FILTer:DEEMphasis:ENABle

Enables or disables the de-emphasis filter for the audio measurement.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:DEEMphasis:ENABle { OFF | ON | 0 | 1 }  
[SENSe]:AUDio:FILTer:DEEMphasis:ENABle?
```

### Arguments

OFF or 0 turns off the de-emphasis filter.

ON or 1 turns on the de-emphasis filter.

### Examples

SENSe:AUDIO:FILTER:DEEMPHASIS:ENABle ON turns on the de-emphasis filter.

## [SENSe]:AUDio:FILTer:DEEMphasis:USER

Sets or queries a user-defined de-emphasis audio filter.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:DEEMphasis:USER <value>  
[SENSe]:AUDio:FILTer:DEEMphasis:USER?
```

### Arguments

<value> ::= <NR3> the value of the user defined filter.

### Examples

SENSe:AUDIO:FILTER:DEEMPHASIS:USER 50.0E-6 sets the user-defined de-emphasis filter to 50  $\mu$ s.

## **[SENSe]:AUDio:FILTer:HPF**

Sets or queries the high-pass filter type.

### **Conditions**

Measurement views: Audio Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:AUDio:FILTer:HPF { H20 | H50 | H300 | H400 | USER }  
[SENSe]:AUDio:FILTer:HPF?
```

### **Arguments**

H20 specifies the 20 Hz filter.

H50 specifies the 50 Hz filter.

H300 specifies the 300 Hz filter.

H400 specifies the 400 Hz filter.

USER specifies a user-defined filter.

### **Examples**

SENSe:AUDIO:FILTER:HPF H300 specifies the 300 Hz filter for the audio measurement.

## **[SENSe]:AUDio:FILTer:HPF:ENABLE**

Enables or disables the high-pass audio filter for the audio measurement.

### **Conditions**

Measurement views: Audio Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:AUDio:FILTer:HPF:ENABLE { OFF | ON | 0 | 1 }  
[SENSe]:AUDio:FILTer:HPF:ENABLE?
```

### **Arguments**

OFF or 0 turns the high-pass audio filter off.

ON or 1 turns the high-pass audio filter on.

## Examples

`SENSe:AUDIO:FILTER:HPF:ENABLE ON` turns the high-pass audio filter on.

## [SENSe]:AUDio:FILTer:HPF:USER

Sets or queries a user-defined high-pass audio filter.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:HPF:USER <value>
```

```
[SENSe]:AUDio:FILTer:HPF:USER?
```

### Arguments

<value> ::= <NR3> the value of the user defined filter.

## Examples

`SENSe:AUDIO:FILTER:HPF:USER 50` sets the user-defined high-pass filter to 50 Hz.

## [SENSe]:AUDio:FILTer:LPF

Sets or queries the low-pass filter type.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:LPF { H300 | H3K | H15K | H30K | H80K | H300K | USER }
```

```
[SENSe]:AUDio:FILTer:LPF?
```

### Arguments

H300 specifies the 300 Hz filter.

H3K specifies the 3 kHz filter.

H15 specifies the 15 kHz filter.

H30K specifies the 30 kHz filter.

H80K specifies the 80 kHz filter.

H300K specifies the 300 kHz filter.

USER specifies a user-defined filter.

## Examples

`SENSe:AUDIO:FILTER:LPF H300` specifies the 300 Hz filter for the audio measurement.

## [SENSe]:AUDio:FILTer:LPF:ENABLE

Enables or disables the low-pass filter for the audio measurement.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:LPF:ENABLE { OFF | ON | 0 | 1 }
```

```
[SENSe]:AUDio:FILTer:LPF:ENABLE?
```

### Arguments

OFF or 0 turns the low-pass audio filter off.

ON or 1 turns the low-pass audio filter on.

## Examples

`SENSe:AUDIO:FILTER:LPF:ENABLE ON` turns the low-pass filter on.

## [SENSe]:AUDio:FILTer:LPF:USER

Sets or queries a user-defined low-pass audio filter.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:LPF:USER <value>
```

```
[SENSe]:AUDio:FILTer:LPF:USER?
```

### Arguments

<value> ::= <NR3> the value of the user defined filter.

### Examples

SENSe:AUDIO:FILTER:LPF:USER 1500 sets the user-defined low-pass filter to 1.50 kHz.

## [SENSe]:AUDio:FILTer:MODE:PREDefined

Selects a predefined audio filter or a filter stored on the instrument file system.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:MODE:PREDefined { OFF | ON | 0 | 1 }
```

```
[SENSe]:AUDio:FILTer:MODE:PREDefined?
```

### Arguments

OFF or 0 specifies the predefined filter on the file system. The file is specified by the [SENSe]:AUDio:FILTer:SOURce? command.

ON or 1 specifies the filter will be one of the selected predefined filters.

### Examples

SENSe:AUDIO:FILTER:MODE:PREDEFINED ON specifies the filter will be one of the predefined filters.

## [SENSe]:AUDio:FILTer:SOURce? (Query Only)

Queries the measurement filter from the user-defined source file.

### Conditions

Measurement views: Audio Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:AUDio:FILTer:SOURce?
```

## Related Commands

[SENSe]:AUDio:FILTer:MODE:PREDefined

## Arguments

None

## Returns

<file name> ::= <string> the name of the predefined filter.

## Examples

SENSe:AUDIO:FILTER:SOURCE? AUD-Test3 loads the predefined filter "AUD-Test3."

## [SENSe]:AUDio:FILTer:STANdard

Sets or queries the Standard audio filter.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:FILTer:STANdard { CCITT | CMSG }
```

```
[SENSe]:AUDio:FILTer:STANdard?
```

## Arguments

CCITT specifies the CCITT Standard filter.

CMSG specifies the CMSG Standard filter.

## Examples

SENSe:AUDIO:FILTER:STANDARD CCITT specifies the CCITT Standard filter.

## [SENSe]:AUDio:FILTer:STANdard:ENABle

Enables or disables a Standard filter for the audio measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands



## Syntax

```
[SENSe]:AUDio:FILTer:STANdard:ENABle { OFF | ON | 0 | 1 }
[SENSe]:AUDio:FILTer:STANdard:ENABle?
```

## Arguments

OFF or 0 turns the standard filter off.

ON or 1 turns the standard audio filter on.

## Examples

SENSe:AUDIO:FILTER:STANDARD:ENABLE ON turns the Standard filter on.

## [SENSe]:AUDio:FREQuency

Sets or queries the reference audio frequency.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:FREQuency <value>
[SENSe]:AUDio:FREQuency?
```

## Arguments

<value> ::= <NR3> the reference audio frequency.

## Examples

SENSe:AUDIO:FREQUENCY? might return 5.824000000E+3 indicating that the reference audio frequency is 5.824 kHz.

## [SENSe]:AUDio:FREQuency:AUTO

Determines whether to set the reference audio frequency automatically or manually.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:FREQuency:AUTO { OFF | ON | 0 | 1 }  
[SENSe]:AUDio:FREQuency:AUTO?
```

## Arguments

OFF or 0 specifies the audio reference frequency is set manually.

ON or 1 specifies the audio reference frequency is set automatically.

## Examples

SENSe:AUDIO:FREQUENCY:AUTO ON sets the audio reference frequency automatically.

## [SENSe]:AUDio:HNOise:ENABLE

Enables or disables the Hum and Noise information in the Audio Summary display.

## Conditions

Measurement views: Audio Summary

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:HNOise:ENABLE { OFF | ON | 0 | 1 }  
[SENSe]:AUDio:HNOise:ENABLE?
```

## Related Commands

```
[SENSe]:AUDio:HNOise:REference
```

## Arguments

OFF or 0 turns off the Hum and Noise information in Audio Summary display.

ON or 1 turns on the Hum and Noise information in the Audio Summary display.

## Examples

SENSe:AUDIO:HNOISE:ENABLE OFF turns off the Hum and Noise information in the Audio Spectrum display.

## [SENSe]:AUDio:HNOise:REference (No Query Form)

Captures the current Hum and Noise information in the Audio Summary display.

## Conditions

Measurement views: Audio Summary

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:HNOise:REFeRence
```

## Related Commands

```
[SENSe]:AUDio:HNOise:ENABle
```

## Arguments

None

## Examples

`SENSe:AUDIO:HNOISE:REFERENCE` shows the current Hum and Noise information in the Audio Summary display.

## [SENSe]:AUDio:SIGnal:TYPE

Sets or queries the audio signal type.

The signal type selection determines the available selections for the `[SENSe]:AUDio:UNITs` command.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:SIGnal:TYPE { AM | FM | DIReCt | PM }
```

```
[SENSe]:AUDio:SIGnal:TYPE?
```

## Related Commands

```
[SENSe]:AUDio:UNITs
```

## Arguments

`AM` sets the signal type to amplitude modulation.

`FM` sets the signal type to frequency modulation.

`DIReCt` sets the signal type to direct modulation.

`PM` sets the signal type to pulse modulation.

## Returns

## Examples

`SENSe:AUDIO:SIGNAL:TYPE FM` sets the signal type to FM, frequency modulation for the audio spectrum measurement.

## [SENSe]:AUDio:SPECTrum:CLEar:RESults (No Query Form)

Clears the results of the audio spectrum measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:SPECTrum:CLEar:RESults
```

## Arguments

None

## [SENSe]:AUDio:SPECTrum:POINts:COUNT

Sets or queries the number of trace points acquired for the audio spectrum display.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:SPECTrum:POINts:COUNT { P801 | P2401 | P4001 | P8001 | P10401 }  
[SENSe]:AUDio:SPECTrum:POINts:COUNT?
```

## Arguments

`P801` sets the number of sample points to 801.

`P2401` sets the number of sample points to 2401.

`P4001` sets the number of sample points to 4001.

`P8001` sets the number of sample points to 8001.

`P10401` sets the number of sample points to 10401.

## Examples

`SENSe:AUDIO:SPECTRUM:POINTS:COUNT P2401` sets the number of sample points to 2401 in the audio spectrum display.

## [SENSe]:AUDio:UNITs

Sets or queries the units of the audio spectrum measurement.

The available selections depend on signal type.

## Conditions

Measurement views: Audio Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:AUDio:UNITs { AM | AM2 | DBAM2 | HZ | HZ2 | DBHZ2 | RAD | RAD2 | DBRAD2 |
DBM | DBV | VOLTs | WATTs }
```

```
[SENSe]:AUDio:UNITs?
```

## Related Commands

```
[SENSe]:AUDio:SIGnal:TYPE
```

## Arguments

The following table lists the arguments.

Argument	Unit	Signal type
DBM	dBm	Direct
DBV	dBV	
VOLTs	Volts	
WATTs	Watts	
AM	am	AM
AM2	am <sup>2</sup>	
DBAM2	dBam <sup>2</sup>	
HZ	Hz	FM
HZ2	Hz <sup>22</sup>	
DBHZ2	dBHz <sup>2</sup>	
RAD	rad	PM
RAD2	rad <sup>2</sup>	
DBRAD2	dBrad <sup>2</sup>	

## Examples

`SENSe:AUDIO:UNITS VOLTs` sets the units to volts for the Direct signal type.

## [SENSe]:AVTime[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Amplitude versus Time for a selected Measurement view number

### Conditions

Measurement views: Amplitude versus Time

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

```
[SENSe]:AVTime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"  
| "Source 4"}
```

```
[SENSe]:AVTime[:MEASview<y>]:SELEcted:SOURce?
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:AVTIME:MEASVIEW1:SELECTED:SOURCE2` queries the Source for Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:SPAN:LINK

Sets or queries the frequency span in the Amplitude versus Time measurement. Programming a specified span disables the `[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}` setting.

### Conditions

Measurement views: Amplitude versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:AVTime[:MEASview<y>]:SPAN:LINK <value>
```

```
[SENSe]:AVTime[:MEASview<y>]:SPAN:LINK?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

OFF or 0 specifies that Link to Span is not checked

ON or 1 specifies that Link to Span is checked

## Examples

`SENSe:AVTIME:MEASVIEW1:SPAN:LINK?` the frequency span link on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}

Sets or queries the time-domain bandwidth filter in the Amplitude versus Time measurement. Programming a specified bandwidth disables the [SENSe]:AVTime[:MEASview<y>]:SPAN setting.

## Conditions

Measurement views: Amplitude versus Time

## Group

Sense commands

## Syntax

[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth} <value>

[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}?

## Arguments

<value>: : <Nrf> specifies the filter bandwidth.

Range: 1 Hz to 20 MHz (Standard) / 60 MHz (Option 110).

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:AVTIME:MEASVIEW1:BANDWIDTH 10MHz` sets the filter bandwidth to 10 MHz on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}:ACTual? (Query Only)

Queries the actual time-domain bandwidth in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}:ACTual?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

<NRf> The actual time-domain bandwidth in Hz.

## Examples

SENSe:AVTIME:MEASVIEW1:BANDWIDTH:ACTUAL? might return 20E+6, indicating that the actual time-domain bandwidth is 20 MHz on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

## Conditions

Measurement views: Amplitude versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:AVTime[:MEASview<y>]:CLEar:RESults
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:AVTIME:MEASVIEW1:CLEAR:RESULTS restarts multi-trace functions on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:MAXTracepoints

Sets or queries the maximum trace points in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time



## Group

Sense commands

## Syntax

```
[SENSe]:AVTime[:MEASview<y>]:MAXTracepoints { ONEK | TENK | HUNDredk |
NEVerdecimate }
```

```
[SENSe]:AVTime[:MEASview<y>]:MAXTracepoints?
```

## Arguments

ONEK sets the maximum trace points to 1 k.

TENK sets the maximum trace points to 10 k.

HUNDredk sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:AVTIME:MEASVIEW1:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:METHod

Sets or queries the method to set the measurement bandwidth in the Amplitude versus Time measurement.

## Conditions

Measurement views: Amplitude versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:AVTime[:MEASview<y>]:METHod { SPAN | TDBW }
```

```
[SENSe]:AVTime[:MEASview<y>]:METHod?
```

## Arguments

SPAN specifies that the measurement bandwidth is set by the frequency span, using the [SENSe]:AVTime:SPAN command.

TDBW specifies that the measurement bandwidth is set by the time-domain bandwidth, using the

```
[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth} command.
```

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:AVTIME:MEASVIEW1:METHOD SPAN` specifies that the measurement bandwidth is set by the frequency span on Amplitude versus Time Measurement View 1.

## [SENSe]:AVTime[:MEASview<y>]:SPAN

Sets or queries the frequency span in the Amplitude versus Time measurement. Programming a specified span disables the `[SENSe]:AVTime[:MEASview<y>]:{BANDwidth|BWIDth}` setting.

### Conditions

Measurement views: Amplitude versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:AVTime[:MEASview<y>]:SPAN <value>
[SENSe]:AVTime[:MEASview<y>]:SPAN?
```

### Arguments

`<value>::<NRf>` specifies the frequency span.

Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

The parameter `<y>=1` to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

### Examples

`SENSe:AVTIME:MEASVIEW1:SPAN 5MHz` sets the frequency span to 5 MHz on Amplitude versus Time Measurement View 1.

## [SENSe]:BLUetooth:ANALysis:LENGth

Sets or returns the analysis length in the Bluetooth displays. Programming a specified length sets `[SENSe]:BLUetooth:ANALysis:LENGth:AUTO` to off.

### Conditions

Measurement views: All Bluetooth displays

### Group

Sense commands

### Syntax

```
[SENSe]:BLUetooth:ANALysis:LENGth <value>
[SENSe]:BLUetooth:ANALysis:LENGth?
```

## Related Commands

[SENSe]:BLUEtooth:ANALysis:LENGth:AUTO

## Arguments

<value> ::= <Nrf> the analysis length. The minimum range depends on the modulation type.

## Examples

SENSe:BLUETOOTH:ANALYSIS:LENGTH 25.625E-6 sets the analysis length to 25.625  $\mu$ s.

## [SENSe]:BLUEtooth:ANALysis:LENGth:ACTual? (Query Only)

Sets the actual analysis length in the Bluetooth displays.

## Conditions

Measurement views: All Bluetooth displays

## Group

Sense commands

## Syntax

[SENSe]:BLUEtooth:ANALysis:LENGth:ACTual?

## Arguments

None

## Returns

<NR3> the actual analysis length in seconds.

## Examples

SENSe:BLUETOOTH:ANALYSIS:LENGTH:ACTUAL? might return 25.625E-6, indicating that the actual analysis length is 25.625  $\mu$ s.

## [SENSe]:BLUEtooth:ANALysis:LENGth:AUTO

Sets or queries the length mode of the Bluetooth displays.

## Conditions

Measurement views: All Bluetooth displays

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:LENGth:AUTO { OFF | ON | 0 | 1 }  
[SENSe]:BLUEtooth:ANALysis:LENGth:AUTO?
```

## Related Commands

```
[SENSe]:BLUEtooth:ANALysis:LENGth
```

## Arguments

OFF or 0 sets the analysis length manually, using the [SENSe]:BLUEtooth:ANALysis:LENGth command.

ON or 1 sets the analysis length automatically.

## Examples

```
SENSe:BLUETOOTH:ANALYSIS:LENGTH:AUTO ON sets the analysis length mode automatically.
```

## [SENSe]:BLUEtooth:ANALysis:MEAS[:BANDwidth|BWIDth]

Sets or returns the measurement bandwidth in the Bluetooth measurement.

## Conditions

Measurement views: All Bluetooth displays

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:MEAS[:BANDwidth|BWIDth] <value>  
[SENSe]:BLUEtooth:ANALysis:MEAS[:BANDwidth|BWIDth]?
```

## Arguments

<value> ::= <NR3>, the measurement bandwidth.

## Examples

```
SENSe:BLUETOOTH:ANALYSIS:MEAS[:BANDWIDTH|BWIDTH]? might return 6.400E+6 indicating the measurement bandwidth is 6.400 MHz.
```

## [SENSe]:BLUEtooth:ANALysis:MEAS:SOURce

Sets or returns the analysis measurement source in the Bluetooth measurement.

## Conditions

Measurement views: All Bluetooth displays

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:MEAS:SOURce { MANual | AUTO | LSP }
```

```
[SENSe]:BLUEtooth:ANALysis:MEAS:SOURce?
```

## Arguments

`MANual` allows for manual selection of the measurement bandwidth. A typical use is setting a narrow measurement bandwidth on a small range of frequencies to examine one signal while rejecting the others that are present in a Spectrum graph.

`AUTO` automatically selects a measurement bandwidth. The measurement picks a bandwidth based on other parameter settings (such as symbol rate and modulation type).

`LSP` sets the measurement bandwidth to Link to Span. This allows you to use the Spectrum display to tune the frequency, view the signal, and then use the Span control to set the measurement bandwidth. This mode emulates legacy instruments.

## Examples

`SENSe:BLUETOOTH:ANALYSIS:MEAS:SOURCE?` might return `MAN` indicating a manual selection of the bandwidth measurement.

## [SENSe]:BLUEtooth:ANALysis:OFFSet

Sets or returns the Bluetooth symbol analysis offset.

## Conditions

Measurement views: Bluetooth displays

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:OFFSet <value>
```

```
[SENSe]:BLUEtooth:ANALysis:OFFSet?
```

## Arguments

`<NR1>` the symbol analysis offset as a percentage. The range is from 0 to 100%.

## Examples

`SENSe:BLUETOOTH:ANALYSIS:OFFSET -50` sets the symbol analysis of set to -50%.

## [SENSe]:BLUEtooth:ANALysis:OFFSet:AUTO

Determines whether to set the carrier frequency error automatically or manually in the Bluetooth displays.

## Conditions

Measurement views: Bluetooth displays

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:OFFSet:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:BLUEtooth:ANALysis:OFFSet:AUTO?
```

## Arguments

OFF or 0 specifies the carrier frequency error is set manually.

ON or 1 specifies the carrier frequency error is set automatically.

## Examples

```
SENSe:BLUETOOTH:ANALYSIS:OFFSET:AUTO ON sets the carrier frequency error automatically.
```

## [SENSe]:BLUEtooth:ANALysis:TIME:UNITs (No Query Form)

Sets the time units for the Bluetooth Constellation settings.

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:ANALysis:TIME:UNITs {SEConds | SYMBols }
```

```
[SENSe]:BLUEtooth:ANALysis:TIME:UNITs?
```

## Arguments

SEConds sets the time units to seconds.

SYMBols sets the time units to symbols.

## Examples

```
SENSe:BLUETOOTH:ANALYSIS:TIME:UNITs sets the time units to seconds.
```

## [SENSe]:BLUEtooth([:BANDwidth]:BWIDth):MODE

Sets or returns the measurement bandwidth mode for the Bluetooth measurements.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth([:BANDwidth]|:BWIDth):MODE { MANual | AUTO |LSP }
[SENSe]:BLUEtooth([:BANDwidth]|:BWIDth):MODE?
```

## Arguments

**MANual** allows for manual selection of the measurement bandwidth. A typical use is setting a narrow measurement bandwidth on a small range of frequencies to examine one signal while rejecting the others that are present in a Spectrum graph.

**AUTO** automatically selects a measurement bandwidth. The measurement picks a bandwidth based on other parameter settings (such as symbol rate and modulation type).

**LSP** sets the Measurement BW to Link to Span. This allows you to use the Spectrum display to tune the frequency, view the signal, and then use the Span control to set the measurement bandwidth. This mode emulates legacy instruments.

## Examples

`SENSe:BLUETOOTH([:BANDWIDTH]|:BWIDTh):MODE AUTO` sets the bandwidth mode to automatically select a measurement bandwidth.

## [SENSe]:BLUEtooth([:BANDwidth]|:BWIDth):TINTerval

Sets or returns the measurement bandwidth frequency span in the Bluetooth displays.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth([:BANDwidth]|:BWIDth):TINTerval <value>
[SENSe]:BLUEtooth([:BANDwidth]|:BWIDth):TINTerval?
```

## Arguments

`<value> ::= <NRf>` specifies the measurement bandwidth in MHz.

## Returns

`<NRf>` the actual measurement bandwidth.

## Examples

`SENSe:BLUETOOTH([:BANDWIDTH]|:BWIDTh):TINTERVAL?` might return `35.255E+6` indicating the bandwidth is 35.255 MHz.

## **[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):TINTerval:AUTO**

Sets or returns the Bluetooth measurement bandwidth to Auto or to Manual.

### **Conditions**

Measurement views: Any Bluetooth display

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):TINTerval:AUTO { 0 | 1 | OFF | ON }  
[SENSe]:BLUEtooth([:BANDwidth]:BWIDth):TINTerval:AUTO?
```

### **Arguments**

OFF or 0 sets measurement bandwidth to Auto.

ON or 1 sets measurement bandwidth to Manual.

### **Examples**

SENSe:BLUETOOTH([:BANDWIDTH]:BWIDTH):TINTERVAL:AUTO? might return 1 indicating the measurement bandwidth is set to Auto.

## **[SENSe]:BLUEtooth:CONStellation:PREFs:GRATicule:SHOW**

Show or hides the graticule state in the Bluetooth Constellation display.

### **Conditions**

Measurement views: Bluetooth Constellation

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:BLUEtooth:CONStellation:PREFs:GRATicule:SHOW { OFF | ON | 0 | 1 }  
[SENSe]:BLUEtooth:CONStellation:PREFs:GRATicule:SHOW?
```

### **Arguments**

OFF or 0 hides the graticule in the display.

ON or 1 shows the graticule in the display.

### **Returns**

0 indicates the graticule is turned off.



1 indicates the graticule is turned on.

## Examples

`SENSe:BLUETOOTH:CONSTELLATION:PREFS:GRATICULE:SHOW ON` turns on the graticule in the display.

## [SENSe]:BLUEtooth:CONStellation:PREFs:MARKERS:SHOW

Show or hides the marker readout in the Bluetooth Constellation display.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:CONStellation:PREFs:MARKERS:SHOW { OFF | ON | 0 | 1 }  
[SENSe]:BLUEtooth:CONStellation:PREFs:MARKERS:SHOW?
```

### Arguments

OFF or 0 hides the marker readout in the display.

ON or 1 shows the marker readout in the display.

### Returns

0 indicates the marker readout is turned off.

1 indicates the marker readout is turned on.

## Examples

`SENSe:BLUETOOTH:CONSTELLATION:PREFS:GRATICULE:SHOW ON` turns on the marker readout in the display.

## [SENSe]:BLUEtooth:CONStellation:PREFs:RADIx

Sets or returns the radix of the marker readout in the Bluetooth Constellation display.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:CONStellation:PREFs:RADIx { Binary | Hex }
```

[SENSe]:BLUEtooth:CONStellation:PREFs:RADIX?

### Arguments

Binary sets the marker readout to Binary.

Hex sets the marker readout to Hex.

### Examples

SENSe:BLUETOOTH:CONSTELLATION:PREFS:RADIX? might return BINARY indicating the Marker readout radix is set to Binary.

## [SENSe]:BLUEtooth:CONStellation:TRACE<x>:CONTent

Sets or returns the appearance of the specified trace (Trace 1 or Trace 2) in the Bluetooth Constellation display.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Sense commands

### Syntax

[SENSe]:BLUEtooth:CONStellation:TRACE<x>:CONTent { Vectors | Points | Lines }

[SENSe]:BLUEtooth:CONStellation:TRACE<x>:CONTent?

### Arguments

Vectors indicates the traces appear as vectors.

Points indicates the traces appear as a series of points.

Lines indicates the traces appear as lines.

### Examples

SENSe:BLUETOOTH:CONSTELLATION:TRACE1:CONTENT VECTORS sets the Trace 1 to show as vectors in the display.

## [SENSe]:BLUEtooth:CONStellation:TRACe<x>:FREEze

Sets or freezes the updates of the specified trace (Trace 1 or Trace 2) in the Bluetooth Constellation display.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:FREEze { OFF | ON | 0 | 1 }
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:FREEze?
```

## Arguments

OFF or 0 normally updates the display of the specified trace.

ON or 1 stops updating the display of the specified trace.

## Examples

`SENSe:BLUETOOTH:CONSTELLATION:TRACE1:FREEZE1` freezes the updates of Trace 1.

## [SENSe]:BLUEtooth:CONStellation:TRACe<x>:POINTsPerSymbol

Sets or returns the number of points per symbol (how many points to use between symbols) for the specified trace (Trace 1 or Trace 2) in the Bluetooth Constellation display.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:POINTsPerSymbol { ONE | TWO | FOUR | EIGHT
| SIXTeen | THIRtytwo }
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:POINTsPerSymbol?
```

## Arguments

The number of points per symbol (1,2,4,8,16,32).

## Examples

`SENSe:BLUETOOTH:CONSTELLATION:TRACE2:POINTSPERSYMBOL FOUR` sets the number of points per symbol for trace 2 to four.

## [SENSe]:BLUEtooth:CONStellation:TRACe<x>:SHOW

Shows or hides the specified trace (Trace 1 or Trace 2) in the Bluetooth Constellation display.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:SHOW { OFF | ON | 0 | 1 }
```

```
[SENSe]:BLUEtooth:CONStellation:TRACe<x>:SHOW?
```

## Arguments

OFF or 0 hides the specified trace in the display.

ON or 1 shows the specified trace in the display.

## Examples

`SENSe:BLUETOOTH:CONSTELLATION:TRACE1:SHOW OFF` hides trace 1 in the display.

## [SENSe]:BLUEtooth:DELTA:AVERAge:FONE

Sets or returns the average frequency for the low deviation pattern in the Bluetooth measurement.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:DELTA:AVERAge:FONE <value>
```

```
[SENSe]:BLUEtooth:DELTA:AVERAge:FONE?
```

## Arguments

`<value>` ::= `<NR3>`, the average frequency of the low deviation pattern.

## Examples

`SENSe:BLUETOOTH:DELTA:AVERAGE:FONE?` might return `139.8309531250E+3` indicating the average frequency of the low deviation pattern is 139.83 kHz.

## [SENSe]:BLUEtooth:DELTA:AVERAge:FTWO

Sets or returns the average frequency for the high deviation pattern in the Bluetooth measurement.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUETOOTH:DELTA:AVERAge:FTWO <value>
```

```
[SENSe]:BLUETOOTH:DELTA:AVERAge:FTWO?
```

## Arguments

<value> ::= <NR3>, the average frequency of the high deviation pattern.

## Examples

SENSe:BLUETOOTH:DELTA:AVERAGE:FTWO? might return 139.8309531250E+3 indicating the average frequency of the low deviation pattern is 139.83 kHz.

## [SENSe]:BLUETOOTH:FDVTime:TRACE:OCTEt? (Query Only)

Returns the selected octet number on the Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Sense commands

## Syntax

```
[SENSe]:BLUETOOTH:FDVTime:TRACE:OCTEt?
```

## Arguments

None

## Returns

<NR3> the octet number.

## Examples

SENSe:BLUETOOTH:FDVTime:TRACE:OCTEt? might return 5 indicating that 5 is the selected octet number.

## [SENSe]:BLUETOOTH:FDVTime:TRACE:OCTEt:TOTAL:COUNT? (Query Only)

Returns the total octet value to be selected on the Frequency Deviation vs. Time display.

For the standard low-energy range the maximum number of octets available is 36. For standard basic rate range, the maximum number of octets depends on the packet type.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:FDVTime:TRACE:OCTEt:TOTAL:COUNT?
```

## Arguments

None

## Returns

An <NR3> number indicating the maximum numbers of octets available.

## Examples

SENSe:BLUEtooth:FDVTime:TRACE:OCTEt:TOTAL:COUNT? might return 36 indicating the maximum number of octets selected is 36.

## [SENSe]:BLUEtooth:FDVTime:TRACE:VIEW:MODE

Sets or queries view mode in the Frequency Deviation vs. Time display.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:FDVTime:TRACE:VIEW:MODE {PACK | OCTET}
```

```
[SENSe]:BLUEtooth:FDVTime:TRACE:VIEW:MODE?
```

## Arguments

PACK indicates the display is in Packet mode.

OCTET indicates the display is in Octet mode.

## Examples

SENSe:BLUEtooth:FDVTime:TRACE:VIEW:MODE PACK sets the view to packet mode.

## [SENSe]:BLUEtooth:FILTer:ALPHa

Sets or returns the alpha filter factor ( $\alpha$ /BT) in the Bluetooth analysis.

### Conditions

Measurement views: Any Bluetooth display

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:FILTer:ALPHa <value>
```

```
[SENSe]:BLUEtooth:FILTer:ALPHa?
```

### Arguments

<value> ::= <NRf> the filter factor; range: 0.001 to 1.

### Examples

SENSe:BLUETOOTH:FILTER:ALPHA 0.5 sets the filter factor to 0.5.

## [SENSe]:BLUEtooth:FILTer:MEASurement

Sets or returns the measurement filter in the Bluetooth measurements.

### Conditions

Measurement views: Any Bluetooth display

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:FILTer:MEASurement {OFF | BR | LE | USER1 | USER2 | UOTH }
```

```
[SENSe]:BLUEtooth:FILTer:MEASurement?
```

### Arguments

OFF no measurement filter is used.

BR, the Basic Rate filter is used.

LE, the Low Energy filter is used.

USER1, the User Meas Filter 1 is used.

USER2, the User Meas Filter 2 is used.

UOTH a user-defined filter is used.

## Examples

`SENSe:BLUETOOTH:FILTER:MEASUREMENT USER1` uses the User Measurement Filter 1 as the measurement filter.

## [SENSe]:BLUEtooth:FILTer:REFerence

Sets or returns the reference filter used in the Bluetooth measurements.

### Conditions

Measurement views: Any Bluetooth display

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:FILTer:REFerence {OFF | BR | USER1 | USER2 | UOTH }
```

```
[SENSe]:BLUEtooth:FILTer:REFerence?
```

### Related Commands

AADD

### Arguments

OFF no reference measurement filter is used.

BR, the Basic Rate filter is used.

USER1, the User Ref Filter 1 is used.

USER2, the User Ref Filter 2 is used.

UOTH, a user-defined filter is used.

### Examples

`SENSe:BLUETOOTH:FILTER:REFERENCE OFF` selects None or no reference filter.

## [SENSe]:BLUEtooth:FREQuency:DEViation

Sets or returns the frequency deviation in the Bluetooth analysis.

### Conditions

Measurement views: Any Bluetooth display

### Group

Sense commands



## Syntax

```
[SENSe]:BLUEtooth:FREQuency:DEVIation <value>
[SENSe]:BLUEtooth:FREQuency:DEVIation?
```

## Related Commands

```
[SENSe]:BLUEtooth:FREQuency:DEVIation:AUTO
```

## Arguments

<value> ::= <NRf>, the frequency deviation.

The Standard product range is 1 Hz to 40 MHz; the Option 110 product range is 1 Hz to 110 MHz.

## Examples

```
SENSe:BLUETOOTH:FREQUENCY:DEVIATION 1MHz sets the frequency deviation to 1 MHz.
```

## [SENSe]:BLUEtooth:FREQuency:DEVIation:AUTO

Determines whether to detect the frequency deviation automatically or manually for the Bluetooth measurement.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:FREQuency:DEVIation:AUTO { 0 | 1 | OFF | ON }
[SENSe]:BLUEtooth:FREQuency:DEVIation:AUTO?
```

## Related Commands

```
[SENSe]:BLUEtooth:FREQuency:DEVIation
```

## Arguments

OFF or 0 sets the frequency deviation manually.

ON or 1 automatically calculates the frequency deviation; (default setting).

## Examples

```
SENSe:BLUETOOTH:FREQUENCY:DEVIATION:AUTO ON automatically calculates the frequency deviation.
```

## [SENSe]:BLUEtooth:FREQuency:ERRor

Sets or returns the frequency error in the Bluetooth analysis parameters (when Auto is off).

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:FREQuency:ERRor <value>
```

```
[SENSe]:BLUEtooth:FREQuency:ERRor?
```

## Related Commands

```
[SENSe]:BLUEtooth:FREQuency:ERRor:AUTO
```

## Arguments

<value> ::= <Nrf>, the frequency error in Hz.

## Examples

`SENSe:BLUETOOTH:FREQUENCY:ERROR 0.02` sets the frequency error to 0.02.

## [SENSe]:BLUEtooth:FREQuency:ERRor:AUTO

Determines whether the frequency error is set to Auto or Manual.

To enter a manual value, when Auto is Off, use the `[SENSe]:BLUEtooth:FREQuency:ERRor` command.

## Conditions

Measurement views: Any Bluetooth display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:FREQuency:ERRor:AUTO { 0 | 1 | OFF | ON }
```

```
[SENSe]:BLUEtooth:FREQuency:ERRor:AUTO?
```

## Arguments

OFF or 0 sets the frequency error manually using the `[SENSe]:BLUEtooth:FREQuency:ERRor` command.

ON or 1 sets the frequency error automatically.

## Examples

`SENSe:BLUETOOTH:FREQUENCY:ERROR:AUTO ON` determines the frequency error automatically.

## [SENSe]:BLUEtooth:INBEmissions:POWEr:LIMIt<x>

Sets or queries the limits of the Bluetooth InBand Emission settings.

When <x> = 1, then commands set or query the  $fTX \pm 2$  MHz limit.

When <x> = 2, then commands set or query the  $fTX \pm (3 + n)$  MHz limit.

### Conditions

Measurement views: Bluetooth InBand Emissions display

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:INBEmissions:POWEr:LIMIt<x> <value>
```

```
[SENSe]:BLUEtooth:INBEmissions:POWEr:LIMIt<x>
```

### Arguments

<value> ::= <NR3> the limit of the specified inband emissions setting in dBm.

### Examples

SENSe:BLUETOOTH:INBEMISSIONS:POWER:LIMIT2? might return -20 indicating the  $fTX \pm 2$  limit is -20 dBm.

## [SENSe]:BLUEtooth:POWEr:CLASs

Sets or queries the power class of the Bluetooth standard.

### Conditions

Measurement views: Bluetooth displays

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:POWEr:CLASs {THREE|TWO|ONE}
```

```
[SENSe]:BLUEtooth:POWEr:CLASs?
```

### Arguments

THREE is power class 3.

TWO is power class 2.

ONE is power class 1.

## Examples

`SENSe:BLUETOOTH:POWER:CLASS THREE` sets the power class to 3.

## [SENSe]:BLUEtooth:STANdard

Sets or queries the Bluetooth standard.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:STANdard {BR|LE|EDR}
```

```
[SENSe]:BLUEtooth:STANdard?
```

### Arguments

BR is the Basic Rate standard.

LE is the Low Energy standard.

EDR is the Eye Diagram standard.

## Examples

`SENSe:BLUETOOTH:STANDARD BR` sets the standard type to Basic Rate.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MAX

Sets or returns the maximum limit of the Basic Rate  $\Delta F1$  average in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MAX<value>
```

```
[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MAX?
```

### Arguments

`<value>` ::= `<NRf>` the maximum average limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:DELTA:FONE:AVERAGE:MAX175.0E+3` sets the maximum Basic Rate limit of the  $\Delta F1$  average to 175.0 kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MIN**

Sets or returns the minimum limit of the Basic Rate  $\Delta F1$  average in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MIN<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FONE:AVERAge:MIN?`

### Arguments

`<value>` ::= `<NRf>` the minimum average limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:DELTA:FONE:AVERAGE:MIN 140.0E+3` sets the minimum limit of the Basic Rate  $\Delta F1$  average to 140.0 kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:PERcent**

Sets or returns the limit of the Basic Rate  $\Delta F2$  percentage in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:PERcent<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:PERcent?`

### Arguments

`<value>` ::= `<NRf>` the limit in percent (%).

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:DELTA:FTWO:PERCENT?` might return 99.00 indicating the limit of the Basic Rate  $\Delta F2$  percentage is 99%.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:RESult**

Sets or returns the limit of the Basic Rate  $\Delta F2$  result in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:RESult<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:DELTA:FTWO:RESult?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:DELTA:FTWO:RESULT 115.0E+3` sets the limit of the Basic Rate  $\Delta F2$  result to 115.0 kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:DRIFt:FNFZero**

Sets or returns the limit of the Basic Rate maximum drift  $f_n - f_0$  in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:DRIFt:FNFZero<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:DRIFt:FNFZero?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:MAX:DRIFT:FNFZERO?` might return `20.00E+3` indicating the limit of the Basic Rate maximum drift  $f_n - f_0$  is 20 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet

Sets or returns the limit of the Basic Rate maximum frequency offset in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:MAX:FREQUENCY:OFFSET 150.0E+3` sets the Basic Rate maximum frequency offset limit to 150 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet:PREAmBDR

Sets or returns the limit of the Basic Rate maximum frequency offset value from the preamble region in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet:PREAmBDR <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:MAX:FREQuency:OFFSet:PREAmBDR?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:MAX:FREQUENCY:OFFSET: PREAMBDR 150.0E+3` sets the limit of the Basic Rate maximum frequency offset limit to 150 kHz.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MAX**

Sets or returns the limit of the Basic Rate Power Class 1 maximum average output power value in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MAX <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MAX?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:ONE:AVERAGE:MAX?` might return 20.00 indicating the maximum output power average is 20.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MIN**

Sets or returns the limit of the Basic Rate Power Class 1 minimum average output power value in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MIN <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:AVERAge:MIN?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.



## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:ONE:AVERAGE:MIN?` might return 0.00 indicating the minimum output power average is 0 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MAX**

Sets or returns the limit of the Basic Rate Power Class 1 maximum peak output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MAX <value>
```

```
[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MAX?
```

## Arguments

<value> ::= <NRf> the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:ONE:PEAK:MAX?` might return 23.00 indicating the maximum peak output power is 23.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MIN**

Sets or returns the limit of the Basic Rate Power Class 1 minimum peak output power value in the Bluetooth Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK:MIN <value>
```

```
[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:ONE:PEAK: MIN?
```

## Arguments

<value> ::= <NRf> the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:ONE:PEAK:MIN?` might return 3.00 indicating the maximum peak output power is 3.00 dBm.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MAX**

Sets or returns the limit of the Basic Rate Power Class 3 maximum average output power value in the Bluetooth Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MAX <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MAX?`

## Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:THREE:AVERAGE:MAX?` might return -40.00 indicating the maximum output power average is -40.00 dBm.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MIN**

Sets or returns the limit of the Basic Rate Power Class 3 minimum average output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MIN <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BDR:OPower:CLASs:THREe:AVERage:MIN?`

## Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:THREE:AVERAGE:MIN?` might return 0.00 indicating the minimum output power average is 0.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MAX**

Sets or returns the limit of the Basic Rate Power Class 3 maximum peak output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MAX <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MAX?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:THREE:PEAK:MAX?` might return 37.00 indicating the maximum peak output power is 37.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MIN**

Sets or returns the limit of the Basic Rate Power Class 3 minimum peak output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MIN <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:THREe:PEAK:MIN?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:THREE:PEAK:MIN?` might return 3.00 indicating the maximum peak output power is 3.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MAX**

Sets or returns the limit of the Basic Rate Power Class 2 maximum average output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MAX <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MAX?`

## Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:TWO:AVERAGE:MAX?` might return 4.00 indicating the maximum output power average is 4.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MIN**

Sets or returns the limit of the Basic Rate Power Class 2 minimum average output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MIN <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:AVERAge:MIN?`

## Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:TWO:AVERAGE:MIN?` might return `-6.00` indicating the minimum output power average is `-6.00` dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MAX**

Sets or returns the limit of the Basic Rate Power Class 2 maximum peak output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MAX <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MAX?`

### Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:TWO:PEAK:MAX?` might return `7.00` indicating the maximum peak output power is `7.00` dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MIN**

Sets or returns the limit of the Basic Rate Power Class 2 minimum peak output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MIN <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BDR:OPOWer:CLASs:TWO:PEAK:MIN?`

### Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BDR:OPOWER:CLASS:TWO:PEAK:MIN?` might return `-3.00` indicating the maximum peak output power is `-3.00` dBm.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MAX**

Sets or returns the maximum limit of the Low Energy  $\Delta F1$  average in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MAX <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MAX?`

### Arguments

`<value>` ::= `<NRf>` the maximum average limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:DELTA:FONE:AVERAGE:MAX 275.0E+3` sets the maximum Low Energy limit of the Low Energy  $\Delta F1$  average to `275.0` kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MIN**

Sets or returns the minimum limit of the Low Energy  $\Delta F1$  average in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MIN <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:DELTA:FONE:AVERAge:MIN?`

### Arguments

`<value>` ::= `<NRf>` the minimum average limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:DELTA:FONE:AVERAGE:MIN 225.0E+3` sets the minimum limit of the Low Energy  $\Delta F1$  average to 225.0 kHz.

## [SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:PERcent

Sets or returns the limit of the Low Energy  $\Delta F2$  percentage in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:PERcent <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:PERcent?`

### Arguments

`<value>` ::= `<NRf>` the limit in percent (%).

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:DELTA:FTWO:PERCENT?` might return 99.00 indicating the limit of the Low Energy  $\Delta F2$  percentage is 99%.

## [SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:RESult

Sets or returns the limit of the Low Energy  $\Delta F2$  result in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:RESult <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:DELTA:FTWO:RESult?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:DELTA:FTWO:RESULT 185.0E+3` sets the limit of the Low Energy  $\Delta F_2$  result to 185.0 kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:DRIFt:FNFZero**

Sets or returns the limit of the Low Energy maximum drift  $f_n - f_0$  in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:DRIFt:FNFZero <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:DRIFt:FNFZero?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:MAX:DRIFT:FNFZERO?` might return `50.0E+3` indicating the limit of the Low Energy maximum drift  $f_n - f_0$  is 50 kHz.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet**

Sets or returns the limit of the Low Energy maximum frequency offset in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.



## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:MAX:FREQUENCY:OFFSET 75.0E+3` sets the Basic Rate maximum frequency offset limit to 75 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet:PREAmble

Sets or returns the limit of the Low Energy maximum frequency offset value from the preamble region in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet:PREAmble <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:MAX:FREQuency:OFFSet:PREAmble?`

### Arguments

`<value> ::= <NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:MAX:FREQUENCY:OFFSET:PREAMBLE75.0E+3` sets the limit of the Low Energy maximum frequency offset limit to 75 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:AVERAge:MAX

Sets or returns the limit of the Low Energy maximum average output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:AVERAge:MAX <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:AVERAge:MAX?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:OPOWER:AVERAGE:MAX?` might return 10.00 indicating the Low Energy maximum output power average is 10.00 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:AVERAge:MIN**

Sets or returns the limit of the Low Energy minimum average output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:AVERAge:MIN <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:AVERAge:MIN?`

## Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:OPOWER:AVERAGE:MIN?` might return -20.00 indicating the Low Energy minimum output power average is -20 dBm.

## **[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:PEAK:MAX**

Sets or returns the limit of the Low Energy maximum peak output power value in the Bluetooth Constellation Summary display.

## Conditions

Measurement views: Bluetooth summary

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:PEAK:MAX <value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:BLE:OPOWer:PEAK:MAX?`

## Arguments

`<value>` ::= `<NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:OPOWER:PEAK:MAX?` might return 13.00 indicating the maximum peak output power is 13.00 dBm.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:PEAK:MIN**

Sets or returns the limit of the Low Energy minimum peak output power value in the Bluetooth Constellation Summary display.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:PEAK:MIN <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:BLE:OPOWer:PEAK:MIN?`

### Arguments

`<value> ::= <NRf>` the limit in dBm.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:BLE:OPOWER:PEAK:MIN?` might return -17.00 indicating the maximum peak output power is -17.00 dBm.

## **[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:F1FZero**

Sets or returns the limit of the maximum  $f_n - f_{n0}$  drift in the packet from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:F1FZero <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:F1FZero?`

### Arguments

`<value> ::= <NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:F1FZERO 20.0E+3` sets the maximum  $f_n$ – $f_{n0}$  drift in the packet to 20.00 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:FNFN5

Sets or returns the limit of the maximum  $f_n$ – $f_{n5}$  drift in the packet from the Bluetooth summary.

### Conditions

Measurement views: Bluetooth summary

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:FNFN5 <value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:FNFN5?`

### Arguments

`<value>` ::= `<NRf>` the limit in Hz.

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:FNFN5 20.0E+3` sets the maximum  $f_n$ – $f_{n5}$  drift in the packet to 20.00 kHz.

## [SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:LECoded:F0FThree

Sets and queries the limit of the LE Coded maximum drift frequency value of F0F3 in the Bluetooth Summary display.

### Conditions

Measurement views: Bluetooth summary display

### Group

Sense commands

### Syntax

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:LECoded:F0FThree <0 | 1>,<value>`

`[SENSe]:BLUEtooth:SUMMary:LIMIt:MAX:DRIFt:LECoded:F0FThree?`

### Arguments

`<value>` ::= `<NRf>`

1 indicates to enable tick mark in “On” column

0 indicates to disable tick mark in “On” column

## Returns

<NRf>

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:LECODED:F0FTHREE 1,85000` sets the limit value to 85 kHz and the tick mark to enabled in the “On” column.

## [SENSe]:BLUEtooth:SUMMArY:LIMIt:MAX:DRIFt:LECoded:F1FFour

Sets and queries the limit of the LE Coded maximum drift frequency value of F1F4 in the Bluetooth Summary display.

## Conditions

Measurement views: Bluetooth summary display

## Group

Sense commands

## Syntax

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:MAX:DRIFt:LECoded:F1FFour <0 | 1>,<value>`

`[SENSe]:BLUEtooth:SUMMArY:LIMIt:MAX:DRIFt:LECoded:F1FFour?`

## Arguments

`<value> ::= <NRf>`

1 indicates to enable tick mark in “On” column

0 indicates to disable tick mark in “On” column

## Returns

<NRf>

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:LECODED:F1FFOUR 1,35000` sets the limit value to 35 kHz and the tick mark to enabled in the “On” column.

## [SENSe]:BLUEtooth:SUMMArY:LIMIt:MAX:DRIFt:LECoded:FNFN3

Sets and queries the limit of the LE Coded maximum drift frequency value of FNFN3 in the Bluetooth Summary display.

## Conditions

Measurement views: Bluetooth summary display

## Group

Sense commands

## Syntax

```
[SENSe]:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:LECODED:FNFN3 <0 | 1>,<value>
```

```
[SENSe]:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:LECODED:FNFN3?
```

## Arguments

<value> ::= <NRf>

1 indicates to enable tick mark in “On” column

0 indicates to disable tick mark in “On” column

## Returns

<NRf>

## Examples

`SENSe:BLUETOOTH:SUMMARY:LIMIT:MAX:DRIFT:LECODED:FNFN3 1,15000` sets the limit value to 15 kHz and the tick mark to enabled in the “On” column.

## [SENSe]:BLUETOOTH:SYMBOL:POINTS

Sets or returns the number of points per symbol (how many points to use between symbols) in the Bluetooth measurement.

## Conditions

Measurement views: Any Bluetooth display.

## Group

Sense commands

## Syntax

```
[SENSe]:BLUETOOTH:SYMBOL:POINTS { ONE | TWO | FOUR | EIGHT | SIXTEEN | THIRTYTWO }
```

```
[SENSe]:BLUETOOTH:SYMBOL:POINTS?
```

## Arguments

The number of points per symbol.

## Examples

`SENSe:BLUETOOTH:SYMBOL:POINTS FOUR` sets the number of points per symbol to four.

## [SENSe]:BLUETOOTH:TEST:PATTERN

Sets or returns the test pattern type in the Bluetooth displays.

## Conditions

Measurement views: Any Bluetooth display.

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:TEST:PATtern
{LDEViation|HDEViation|TPOTHER}
[SENSe]:BLUEtooth:TEST:PATtern?
```

## Related Commands

```
[SENSe]:BLUEtooth:TEST:PATtern:AUTO:DETEct
```

## Arguments

LDEViation indicates the low deviation test pattern.

HDEViation indicates the high deviation test pattern.

TPOTHER indicates a user defined test pattern.

## Examples

SENSe:BLUETOOTH:TEST:PATTERN? might return LDEV indicating the selected test pattern type is the low deviation type.

## [SENSe]:BLUEtooth:TEST:PATtern:AUTO:DETEct

Sets or returns the auto detect setting for the Bluetooth displays.

## Conditions

Measurement views: Any Bluetooth display.

## Group

Sense commands

## Syntax

```
[SENSe]:BLUEtooth:TEST:PATtern:AUTO:DETEct {0|1}
[SENSe]:BLUEtooth:TEST:PATtern:AUTO:DETEct?
```

## Related Commands

```
[SENSe]:BLUEtooth:TEST:PATtern
```

## Arguments

1 indicates that the auto detect test pattern is enabled (checked in the view). 0 indicates that the auto detect test pattern is not enabled (not checked).

## Examples

`SENSe:BLUETOOTH:TEST:PATTERN:AUTO:DETECT?` might return 0 indicating that the auto detect test pattern is not enabled.

## [SENSe]:BLUEtooth:TIME:ZERO:REFERENCE

Sets or returns the Time Zero Reference setting in the Bluetooth display.

### Conditions

Measurement views: Any Bluetooth display.

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:TIME:ZERO:REFERENCE {Acquisistion Start|Trigger}
```

```
[SENSe]:BLUEtooth:TIME:ZERO:REFERENCE?
```

### Arguments

Acquisition Start sets the Time Zero Reference to the start of the acquisition.

Trigger sets the Time Zero Reference to the trigger position.

## Examples

`SENSe:BLUETOOTH:TIME:ZERO:REFERENCE?` might return TRIGGER indicating the Time Zero Reference is the trigger position.

## [SENSe]:BLUEtooth:TRACe:TYPE

Sets or queries the type of trace on the Trace tab of the control panel.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:BLUEtooth:TRACe:TYPE {IQ|FREQDEV}
```

```
[SENSe]:BLUEtooth:TRACe:TYPE?
```

### Arguments

IQ specifies the IQ trace.



FREQDEV specifies the Frequency Deviation trace.

## Examples

`SENSe:BLUETOOTH:TRACE:TYPE?` might return IQ indicating the Trace type is set to IQ.

## [SENSe]:CONSte[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Constellation for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (ex: GPDemod Constellation).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:CONSte[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"
| "Source 4"}
```

```
[SENSe]:CONSte[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

`SENSe:CONSte:MEASview1:SELEcted:SOURce?` queries the Source for Constellation Measurement View 1.

## [SENSe]:CCDF:{BANDwidth|BWIDth}

Sets or queries the CCDF measurement bandwidth (frequency span).

## Conditions

Measurement views: CCDF

## Group

Sense commands

## Syntax

```
[SENSe]:CCDF:{BANDwidth|BWIDth} <value>
```

```
[SENSe]:CCDF:{BANDwidth|BWIDth}?
```

## Arguments

<value> ::= <NRf> is the CCDF measurement bandwidth.

Range: 10 Hz to 40 MHz (Standard) / 60 MHz (Option 110).

## Examples

SENSe:CCDF:BANDWIDTH 1MHz sets the CCDF measurement bandwidth to 1 MHz.

## [SENSe]:CCDF:CLEAr (No Query Form)

Clears the CCDF accumulator and restarts the measurement.

## Conditions

Measurement views: CCDF

## Group

Sense commands

## Syntax

```
[SENSe]:CCDF:CLEAr
```

## Arguments

None

## Examples

SENSe:CCDF:CLEAr clears the CCDF accumulator and restarts the measurement.

## [SENSe]:CCDF:TIME:TOTAl:LENGth

Sets or queries the CCDF measurement time when [SENSe]:CCDF:TIME:TYPE is set to TOTAl.

## Conditions

Measurement views: CCDF

## Group

Sense commands

## Syntax

```
[SENSe]:CCDF:TIME:TOTAl:LENGth <value>
```

```
[SENSe]:CCDF:TIME:TOTAL:LENGTH?
```

## Arguments

<value>::<NRf> specifies the CCDF measurement time.

Range: 20 ms to 100 s.

## Examples

SENSe:CCDF:TIME:TOTAL:LENGTH 10 sets the CCDF measurement time to 10 s.

## [SENSe]:CCDF:TIME:TYPE

Determines how to repeat the CCDF measurement.

## Conditions

Measurement views: CCDF

## Group

Sense commands

## Syntax

```
[SENSe]:CCDF:TIME:TYPE { SINGLE | TOTAL | CONTInuous }
```

```
[SENSe]:CCDF:TIME:TYPE?
```

## Related Commands

INITiate commands

## Arguments

SINGLE specifies that the analyzer sets the analysis length to 1 ms and then acquire data once to calculate CCDF.

TOTAL specifies that the analyzer sets the analysis length to 20 ms and then repeats data acquisition and CCDF calculation for the time specified by the [SENSe]:CCDF:TIME:TOTAL:LENGTH command.

CONTInuous specifies that the analyzer sets the analysis length to 1 ms and then repeats data acquisition and CCDF calculation continuously. To reset the process, use the [SENSe]:CCDF:CLEAr command or the INITiate commands.

## Examples

SENSe:CCDF:TIME:TYPE SINGLE specifies that the analyzer sets the analysis length to 1 ms and then acquire data once to calculate CCDF.

## [SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth

Sets or queries the analysis length in seconds or symbols. The command [SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs determines which is used. For example, if the symbol rate is 1 MHz and the acquisition length is 20 ms, the range may be 0 to 19999. Setting a specified length changes [SENSe]:DDEMod[:SOURce<x>]:ANALysis:AUTO to OFF.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:ANALYSIS:LENGTh<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:ANALYSIS:LENGTh?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:ANALYSIS:LENGTh:ACTual?](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:TIME:UNITs](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:ANALYSIS:LENGTh:AUTO](#)

## Arguments

<x> = 1 to 8, representing Range of Source

<value>::<NRf> specifies the analysis length in seconds or symbols.

Range in seconds: 200 ns to [(acquisition length) - 400 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns].

Range in symbols: 200 ns \* (symbol rate) to [(acquisition length) - 400 ns] \* (symbol rate).

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual analysis length is reduced to [(acquisition length) - 200 ns] \* (symbol rate).

## Examples

`SENSe:DDEMOD:SOURce1:ANALYSIS:LENGTh 25.625µs` sets the analysis length of the measurement source1 to 25.625 µs.

`SENSe:DDEMOD:SOURce1:ANALYSIS:LENGTh 256` sets the analysis length of the measurement source1 to 256 symbols.

## **[SENSe]:DDEMod[:SOURce<x>]:ANALYSIS:LENGTh:ACTual? (Query only)**

Queries the actual analysis length which returns a value in either seconds or symbols. The command

`[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs` determines which is used.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:ACTual?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:ANALysis:LENGth](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:TIME:UNITs](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

## Returns

<NRf> Actual analysis length in seconds or symbols.

## Examples

`SENSe:DDEMOD:SOURCE1:ANALYSIS:LENGTH:ACTUAL?` might return 25.625E-6, indicating that the actual analysis length of the measurement source1 is 25.625  $\mu$ s.

## [SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:AUTO

Determines whether to set the analysis length of the measurement source automatically or manually.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth:AUTO?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 sets the analysis length manually, using the `[SENSe]:DDEMod[:SOURce<x>]:ANALysis:LENGth` command.

ON or 1 sets the analysis length automatically.

## Examples

`SENSe:DDEMOD:SOURce1:ANALYSIS:LENGTH:AUTO ON` sets the analysis length of the measurement source1 automatically.

## **[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:MODE**

Sets or queries the measurement bandwidth mode of the measurement source for Digital Demod measurements.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:MODE { MANual | AUTO | LSPan }
```

```
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:MODE?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:{BANDwidth|BWIDth}:TINterval](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

`MANual` specifies manual bandwidth control mode. A typical use is setting a narrow measurement bandwidth on a small range of frequencies to examine one signal while rejecting the others that are present in a Spectrum graph.

`AUTO` specifies automatic bandwidth control mode where a measurement picks a bandwidth based on other parameter settings (such as symbol rate, modulation type, filter, and so on).

`LSPan` (LinkToSpan) specifies Lspan bandwidth control mode, where you can use the Spectrum display to tune the frequency, view the signal, and use the Span control to set the measurement bandwidth. This mode emulates legacy instruments.

## Examples

`SENSe:DDEMOD:SOURce1:BANDWIDTH:MODE AUTO` sets the analyzer bandwidth mode of the measurement source1 to automatically select a measurement bandwidth.

## **[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINterval**

Sets or queries the measurement bandwidth of the measurement source (frequency span).

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINTerval<value>
```

```
[SENSe]:[:SOURce<x>]:{BANDwidth|BWIDth}:TINTerval?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SRATe](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

<value> ::= <NRf> specifies the measurement bandwidth.

Range: Symbol rate to 40 MHz or with option 110, 110 MHz.

The minimum value depends on the setting of `[ :SENSe ] :DDEMod [ :SOURce<x> ] :SRATe`.

## Returns

<NRf> Actual analysis length in seconds or symbols.

## Examples

`SENSe:DDEMOD:SOURce1:BANDWIDTH:TINTERVAL 35.255MHz` sets the measurement bandwidth of the measurement source1 to 35.255 MHz.

## [SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINTerval:AUTO

Sets or queries the measurement bandwidth of the measurement source (frequency span) automatically.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINTerval:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINTerval:AUTO?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:{BANDwidth|BWIDth}:TINTerval](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 sets the measurement bandwidth manually, using the `[SENSe]:DDEMod[:SOURce<x>]:{BANDwidth|BWIDth}:TINTERval` command.

ON or 1 sets the measurement bandwidth automatically.

## Examples

`SENSe:DDEMOD:SOURce1:BANDWIDTH:TINTERVAL:AUTO` sets the measurement bandwidth of the measurement source1 automatically.

## [SENSe]:DDEMod[:SOURce<x>]:BURSt:DETECT

Sets or queries burst detection of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

`[SENSe]:DDEMod[:SOURce<x>]:BURSt:DETECT { ON | OFF }`

`[SENSe]:DDEMod[:SOURce<x>]:BURSt:DETECT?`

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:BURSt:THReshold](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

ON analyzes just that burst period if a burst is found. If a burst is not found, does not analyze but displays an error message.

OFF analyzes the whole analysis length.



**Note:** When selecting On and if the signal is not adequate for the demodulation, the measurement will fail and show an error message.

## Examples

`SENSe:DDEMOD:SOURce1:BURST:DETECT OFF` disables burst detection of the measurement source and analyzes the whole analysis length.



## [SENSe]:DDEMod[:SOURce<x>]:BURSt:THReshold

Sets or queries the threshold level of the measurement source above which the input signal is determined to be a burst.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:BURSt:THReshold<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:BURSt:THReshold?
```

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:BURSt:DETECT](#)

### Arguments

<x> = 1 to 8, representing Range of Source

<value>::<NRf> specifies the threshold level for detecting bursts.

Range: -100 to -10 dBc.

### Examples

SENSe:DDEMOD:SOURce1:BURST:THRESHOLD -25 sets the threshold level of the measurement source1 to -25 dBc.

## [SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet

Sets or queries the carrier frequency offset for the measurement source in digital modulation analysis.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet?
```

## Related Commands

The settings of the following commands may narrow the effective range.

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:ALPHa](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:FILTer:REFerence](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:MODulation:TYPE](#)

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SRATe](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

<value> ::= <NRf> specifies the carrier frequency offset.

Range: -20 MHz to +20 MHz or with Option 110, -55 MHz to +55 MHz.

## Examples

`SENSe:DDEMOD:SOURCE1:CARRIER:OFFSET 2kHz` sets the carrier frequency offset to 2 kHz for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet:AUTO

Sets or queries whether to detect the carrier frequency automatically for the measurement source in digital modulational analysis.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet:AUTO?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:CARRier:OFFSet](#)

## Arguments

<x> = 1 to 8, representing Range of Source

OFF or 0 sets the carrier frequency manually, using the `[SENSe]:DDEMod[:SOURce<x>]:CARRier:OFFSet` command.

ON or 1 detects the carrier frequency automatically.

## Examples

`SENSe:DDEMOD:SOURce1:CARRIER:OFFSET:AUTO ON` enables automatic detection of the carrier frequency for measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:AVAIlable? (Query only)

Determines if the equalizer is available for the measurement source.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:AVAIlable?
```

### Arguments

<x> = 1 to 8, representing Range of Source.

### Returns

0 indicates the equalizer is not available for the current modulation type and reference filter.

1 indicates the equalizer is available.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:CONVergence

Sets or queries the Convergence value (also known as tap update rate) of the measurement source.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:CONVergence <value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:CONVergence?
```

### Arguments

<x> = 1 to 8, representing Range of Source.

<value>::<NR3> specifies the convergence value.

## Examples

SENSe:DDEMOD:SOURce1:EQUALIZER:CONVERGENCE? might return 1.000000000E-6 for measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:ENABle

Sets or queries the Equalizer state of the measurement source.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:ENABle {OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:ENABle?
```

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:EQUalizer:MODE](#)

### Arguments

<x> = 1 to 8, representing Range of Source.

ON or 1 enables the Equalizer.

OFF or 0 disables the Equalizer.

### Examples

SENSe:DDEMod:SOURce1:EQU:ENAB 1 might return 1, indicating that the equalizer is enabled for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:LENGth

Sets or queries the equalizer filter length of measurement source.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:LENGth<value>
```

## Arguments

<x> = 1 to 8, representing Range of Source.

<value>::<NR1> specifies the equalizer filter length.

## Examples

SENSe:DDEMOD:SOURce1:EQUALIZER:LENGTH 39 sets the filter length of measurement source1 to 39 symbols.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:MODE

Sets or queries the equalizer mode of the measurement source. The equalizer can in learning mode (train) or enabled (hold).

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:MODE {TRAIN | HOLD }
```

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:MODE?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:EQUalizer:ENABle](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

TRAIN places the equalizer in training mode.

HOLD places in the equalizer in enabled mode.

## Examples

SENSe:DDEMOD:SOURce1:EQUALIZER:MODE HOLD places the equalizer in enabled mode for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:RESet (No query form)

Resets the equalizer settings of the measurement source to default values.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:RESet
```

## Examples

SENSe:DDEMOD:SOURce1:EQUALIZER:RESET sets the equalizer parameters of the measurement source1 to their default values.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TAPS

Sets or queries the number of filter coefficients for the equalizer of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TAPS <value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TAPS?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

<value>::=<NR1> the filter coefficient for the equalizer.

## Examples

SENSe:DDEMOD:SOURce<x>:EQUALIZER:TAPS 65 sets the measurement source1 filter coefficient to 65.

## [SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TSRAtio

Sets or queries the Equalizer taps or Symbol parameter for the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:EQUalizer:TSRatIo { ONE | TWO | FOUR | EIGHT }
```

## Arguments

Valid arguments are ONE, TWO, FOUR, or EIGHT. Numerals are not valid arguments.

<x> = 1 to 8, representing Range of Source.

## Examples

SENSe:DDEMOD:SOURCE1:EQUALIZER:TSRATIO FOUR sets the Taps or Symbol value to four for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:FILTer:ALPHa

Sets or queries the filter factor ( $\alpha$ /BT) for the measurement source in the digital modulation analysis.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:ALPHa <value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:ALPHa?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

<value>::<NRF> specifies the filter factor. Range: 0.001 to 1.

## Examples

SENSe:DDEMOD:SOURCE1:FILTER:ALPHA 0.5 sets the filter factor to 0.5 for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:FILTer:MEASurement

Sets or queries the measurement filter for the measurement source in the digital modulation analysis.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:MEASurement { OFF | RRCosine | RCOSine | GAUSSian | RECTangular | IS95TXEQ_MEA | IS95TX_MEA | C4FM_P25 | USERx | UOTHer }
```

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:MEASurement?
```

## Arguments

The following table lists the arguments.

<x> = 1 to 8, representing Range of Source.

**Table 45: Digital modulation measurement filter**

Argument	Measurement filter
OFF	No filter
RRCosine	Root Raised Cosine
RCOSine	Raised Cosine
GAUSSian	Gaussian
RECTangular	Rectangular
IS95TXEQ_MEA	IS95 receive filter for the transmitter configured with both the transmit filter and the phase equalizer.
IS95TX_MEA	IS95 receive filter for the transmitter configured with only the transmit filter.
C4FM_P25	C4FM-P25
USER1	User defined Measurement Filter 1
USER2	User defined Measurement Filter 2
USER3	User defined Measurement Filter 3
UOTHer	Other user defined Measurement Filter

## Examples

`SENSe:DDEMOD:SOURCE1:FILTer:MEASUREMENT RRCosine` selects the Root Raised Cosine filter as the measurement filter for the measurement source1.

## [SENSe]:DDEMod:SOURCE<x>:FILTer:REFerence

Sets or queries the reference filter for the measurement source in the digital modulation analysis..



## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:REFEreNce { OFF | RCOSine | GAUSSian |
RECTangular | IS95REF | HSINe | SOQPSK_MIL | SOQPSK_ARTM | SBPSK_MIL | USERx |
UOTHer }
```

```
[SENSe]:DDEMod[:SOURce<x>]:FILTer:REFEreNce?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

The following table lists the arguments.

**Table 46: Digital modulation reference filter**

Argument	Measurement filter
OFF	No filter
RCOSine	Raised Cosine
GAUSSian	Gaussian
RECTangular	Rectangular
IS95REF	IS95 reference filter including the response of the transmit filter, the phase equalizer, and the receive (complementary) filter.
HSINe	Half Sine
SOQPSK_MIL	SOQPSK-MIL
SOQPSK_ARTM	SOQPSK-ARTM
SBPSK_MIL	SBPSK-MIL
USER1	User defined Measurement Filter 1
USER2	User defined Measurement Filter 2
USER3	User defined Measurement Filter 3
UOTHer	Other user defined Measurement Filter

## Examples

`SENSe:DDEMOD:SOURCE1:FILTER:REFERENCE RCOSine` selects the Raised Cosine filter as the reference filter for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEViation

Sets or queries the frequency deviation of the measurement source in the digital modulation analysis. Programming a specified frequency deviation sets `[SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEViation:AUTO OFF`.

This command is valid when `[SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE` is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation?
```

### Arguments

<x> = 1 to 8, representing Range of Source.

<value>::<NRf> sets the frequency deviation.

Standard product range: 1 Hz to 40 MHz.

Option 110 product range: 1 Hz to 110 MHz.

### Examples

`SENSe:DDEMOD:SOURce1:FREQUENCY:DEVIATION 1MHz` sets the frequency deviation of the measurement source1 to 1 MHz.

## [SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation:AUTO

Determines whether to detect automatically or set manually the frequency deviation of the measurement source used to determine the symbol values of an FSK or C4FM signal.

This command is valid when `[SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE` is set to C4FM, FSK2, FSK4, FSK8, or FSK16.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation:AUTO?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

ON or 1 automatically calculates the frequency deviation for the analysis range (default).

OFF or 0 sets the frequency deviation using the [SENSe]:DDEMod[:SOURce<x>]:FREQuency:DEVIation command.

## Examples

SENSe:DDEMOD:SOURce1:FREQUENCY:DEVIATION:AUTO ON automatically calculates the frequency deviation of the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:MAGNitude:NORMALize

Sets or queries the method for the magnitude normalization of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:MAGNitude:NORMALize { RSYMBOL | MSYMBOL }
[SENSe]:DDEMod[:SOURce<x>]:MAGNitude:NORMALize?
```

## Arguments

<x> = 1 to 8, representing Range of Source

RSYMBOL normalizes the magnitude with the RMS symbol magnitude.

MSYMBOL normalizes the magnitude with the maximum symbol magnitude.

## Examples

SENSe:DDEMOD:SOURce1:MAGNITUDE:NORMALIZE RSYMBOL normalizes the magnitude of measurement source1 with the RMS symbol magnitude.

## [SENSe]:DDEMod[:SOURce<x>]:MINDex

Sets or queries the measurement source modulation index of a CPM signal.

This command is valid when [SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE is set to CPM and [SENSe]:DDEMod[:SOURce<x>]:MINDex:AUTO is set to OFF.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:MINDEX { 1 | 2 | 3 | 4 | 5 | 6 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:MINDEX?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

The following table shows the arguments and modulation index.

**Table 47: CPM modulation index**

Argument	Modulation index
1	4/16, 5/16
2	5/16, 6/16
3	6/16, 7/16
4	7/16, 10/16
5	12/16, 13/16
6	8/16, 8/16

## Examples

SENSe:DDEMOD:SOURCE1:MINDEX 1 selects the measurement source1 modulation index to "4/16, 5/16".

## [SENSe]:DDEMod[:SOURce<x>]:MINDEX:AUTO

Determines whether to detect automatically or set manually the measurement source modulation index of a CPM signal. This command is valid when [SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE is set to CPM.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:MINDEX:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:MINDEX:AUTO
```

## Arguments

<x> = 1 to 8, representing Range of Source.

ON or 1 automatically calculates the modulation index for the analysis range.

OFF or 0 sets the modulation index using the [SENSe]:DDEMod[:SOURce<x>]:MINDEX command.

## Examples

SENSe:DDEMOD:SOURce1:MINDEX:AUTO ON automatically calculates the measurement source1 modulation index for the analysis range.

## [SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE

Sets or queries the modulation type for the measurement source in the digital modulation analysis.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE { QPSK | PSK8 | D8PSK | PIOVER2DBPSK |
DQPSK | PIOVER4DQPSK | BPSK | OQPSK | QAM16 | QAM32 | QAM64 | QAM128 | QAM256 |
QAM1024 | MSK | FSK2 | FSK4 | FSK8 | FSK16 | CPM | APSK16| APSK32 | SOQPSK | SBPSK
| C4FM }
```

```
[SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

The following table lists the arguments and corresponding modulation type.

**Table 48: Modulation type**

Argument	Modulation type
QPSK	QPSK
PSK8	8PSK
D8PSK	D8PSK
PIOVER2DBPSK	$\pi/2$ DBPSK
DQPSK	DQPSK
PIOVER4DQPSK	$\pi/4$ QPSK
BPSK	BPSK
OQPSK	OQPSK

Table continued...

Argument	Modulation type
QAM16	16QAM
QAM32	32QAM
QAM64	64QAM
QAM128	128QAM
QAM256	256QAM
QAM1024	1024QAM
MSK	MSK
FSK2	FSK2
FSK4	FSK4
FSK8	FSK8
FSK16	FSK16
CPM	CPM
APSK16	APSK16
APSK32	APSK32
SOQPSK	SOQPSK
SBPSK	SBPSK
C4FM	C4FM

## Examples

`SENSe:DDEMOD:SOURCE1:MODULATION:TYPE QPSK` selects QPSK modulation system for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:PRESet (No query form)

Presets the modulation analysis analysis of the measurement source to a communication standard.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

`[SENSe]:DDEMod[:SOURce<x>]:PRESet <standard_name>`

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:MODulation:TYPE](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

The following table lists the preset standard names with their modulation types and settings.

**Table 49: Modulation Presets**

Preset Name	Modulation type	Settings
"802.15.4"	OQPSK	Symbol rate: 1 MHz
		Meas Filter: None
		Ref Filter: Half sine
		Filter Param: None
		Symbol Pt Location: NA
		Remove Q offset: not checked
"SBPSK-MIL"	SBPSK	Symbol rate: 2.4 kHz
		Meas Filter: None
		Ref Filter: SBPSK-MIL
		Filter Param: 0.5
		Symbol Pt Location: NA
		Remove Q offset: NA
"SOQPSK-MIL"	SOQPSK	Symbol rate: 2.4 kHz
		Meas Filter: None
		Ref Filter: SOQPSK-MIL
		Filter Param: 0.5
		Symbol Pt Location: Center
		Remove Q offset: checked
"CPM-MIL"	CPM	Symbol rate: 19.2 KHz
		Meas Filter: None
		Ref Filter: None
		Filter Param: None
		Symbol Pt Location: NA
		Remove Q offset: NA
"SOQPSK-ARTM Tier 1"	SOQPSK	Symbol rate: 2.5 MHz
		Meas Filter: None
		Ref Filter: SOQPSK-ARTM
		Filter Param: None
		Symbol Pt Location: Center
		Remove Q offset: checked

Table continued...

Preset Name	Modulation type	Settings
"Project25 Phase I"	C4FM	Symbol rate: 4.8 kHz
		Meas Filter: C4FM-P25
		Ref Filter: Raised cosine
		Filter Param: 0.2
		Symbol Pt Location: NA
		Remove Q offset: NA
"CDMA2000-Base"	QPSK	Symbol rate: 1.2288 MHz
		Meas Filter: IS95 TXEQ_MEA
		Ref Filter: IS95 REF
		Filter Param: None
		Symbol Pt Location: NA
		Remove Q offset: NA
"W-CDMA"	QPSK	Symbol rate: 3.84 MHz
		Meas Filter: Root raised cosine
		Ref Filter: Raised cosine
		Filter Param: 0.22
		Symbol Pt Location: NA
		Remove Q offset: NA
In addition to the specific settings listed in the table, the following general settings are also made when you load any of the defined presets.		

Setting	Value
Points/symbol	4
Burst detection mode	Off
Burst detection threshold	-10 dBc
Analysis offset	Auto
Analysis length	Auto
Frequency offset	Auto
Measurement BW	Auto
Frequency deviation	Auto
Modulation index	Auto

### Examples

`SENSe:DDEMOD:SOURCE1:PRESET "802.15.4"` the measurement source1 demodulation to the standard QPSK modulation.

### **[SENSe]:DDEMod[:SOURce<x>]:RING:RADIus**

Sets or queries the ring radius of the measurement source in the digital APSK modulation analysis.



## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:RING:RADIus<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:RING:RADIus?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

<value> : :<NRf> specifies the ring radius.

APSK16 valid ring arguments are 1 and 2.

APSK32 valid ring arguments are 1, 2, and 3.

## Examples

```
SENSe:DDEMOD:SOURce1:RING:RADIUS
```

## [SENSe]:DDEMod[:SOURce<x>]:RING:ROTation

Sets or queries the ring rotation of the measurement source in the digital APSK modulation analysis.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:RING:ROTation<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:RING:ROTation?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

<value> : :<NRf> specifies the ring rotation.

APSK16 valid ring arguments are 1 and 2.

APSK32 valid ring arguments are 1, 2, and 3.

## Examples

```
SENSe:DDEMOD[:SOURce<x>]:RING:ROTATION
```

## [SENSe]:DDEMod[:SOURce<x>]:SRATe

Sets or queries the symbol rate for the measurement source in the digital modulation analysis.

### Conditions

Measurement views: General purpose digital modulation

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SRATe<value>
```

```
[SENSe]:DDEMod[:SOURce<x>]:SRATe?
```

### Arguments

<x> = 1 to 8, representing Range of Source.

<value>: :<NRf> specifies the symbol rate.

Standard product range: 100 Hz to 40 MHz.

Option 110 product range: 100 Hz to 110 MHz.

### Examples

```
SENSe:DDEMOD:SOURCE1:SRATE 21.0E3 sets the symbol rate of measurement source1 to 21 kHz.
```

## [SENSe]:DDEMod[:SOURce<x>]:SWAP:IQ

Determines whether or not to exchange I and Q data of the measurement source before demodulating.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SWAP:IQ { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SWAP:IQ?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 uses I and Q data as they are.

ON or 1 exchanges I and Q data.

## Examples

SENSe:DDEMOD:SOURce1:SWAP:IQ ON exchanges I and Q data of the measurement source1 before demodulating.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:HSSHift

Sets or queries the measurement source Q data half-symbol shift for OQPSK and SOQPSK signals.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:HSSHift { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:HSSHift?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:MODulation:TYPE](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 the Q offset shift is not applied.

ON or 1 the Q offset shift is applied.

## Examples

SENSe:DDEMod:SOURce1:SYMBOL:HSSHift ON the measurement source Q offset is applied.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP:SOURce? (Query only)

Queries the user symbol map of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP:SOURce?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

## Examples

SENSe:DDEMOD:SOURCE1:SYMBOL:MAP:SOURCE? returns the symbol map filename of the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP[:STATe]

Determines whether or not to use the user symbol map of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP[:STATe] { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:MAP[:STATe] ?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SYMBOL:MAP:SOURce<x>](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 disables the user symbol map.

ON or 1 enables the user symbol map.

## Examples

SENSe:DDEMOD:SOURCE1:SYMBOL:MAP:STATE ON enables the user symbol map of measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:PLOT:POStion

Sets or queries the symbol point location for measurement source on an SOQPSK waveform.

## Conditions

Source keyword is optional, if not present Source1 will be default source.

Effective on SOQPSK waveform and any general purpose digital demodulation views except Signal Quality and Symbol Table.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:PLOT:POSITION { EDGE | MIDDLE }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:PLOT:POSITION?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

EDGE sets the symbol location at the edge of the eye opening.

MIDDLE sets the symbol location at the middle of the eye opening.

## Examples

SENSe:DDEMOD:SOURCE1:SYMBOL:PLOT:POSITION EDGE sets the symbol location the edge of the eye opening for the measurement source 1.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:POINTS

Sets or queries the number of points per symbol of the measurement source (how many points to use between symbols when connecting the dots).



**Note:** 1 is not valid for the GMSK modulation.

In the constellation view, select VECTors using the TRACe<x>:CONStE:MODE command first to change Points/Symbol.

## Conditions

Measurement views: Constellation, EVM versus Time, Magnitude error versus Time, Phase error versus Time, Freq Dev versus Time, Demod I&Q versus Time.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:POINTS { ONE | TWO | FOUR | EIGHT }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:POINTS?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

ONE, TWO, FOUR, and EIGHT represent the number of points per symbol.

The following table lists the conditions of use.

Argument	Supported modulation type
ONE	All except SOQPSK, OQPSK, SBPSK
TWO	All except SOQPSK
THREE	All
FOUR	All

## Examples

`SENSe:DDEMOD:SOURCE1:SYMBOL:POINTS FOUR` sets the number of points per symbol of the measurement source1 to four.

## [SENSe]:DDEMod[:SOURce<x>]:SYMBOL:RATE:SEARCh

Determines whether to enable a symbol rate search of the measurement source. This command is valid when the demodulation type is set to FSK2, FSK4, FSK8, or FSK16.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

### Group

Sense commands

### Syntax

`[SENSe]:DDEMod[:SOURce<x>]:SYMBOL:RATE:SEARCh { OFF | ON | 0 | 1 }`

### Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:MODulation:TYPE](#)

### Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 disables the search.

ON or 1 enables the search.

### Examples

`SENSe:DDEMOD:SOURCE1:SYMBOL:RATE:SEARCH ON` enables the symbol rate search of measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD

Determines whether to enable the synchronization word of the measurement source.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD { OFF | ON | 0 | 1 }
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD?
```

## Related Commands

[\[SENSe\]:DDEMod\[:SOURce<x>\]:SYNCh:WORD:SYMBOL](#)

## Arguments

<x> = 1 to 8, representing Range of Source.

OFF or 0 disables the synchronization word.

ON or 1 enables the synchronization word.

## Examples

SENSe:DDEMOD:SOURCE1:SYNCH:WORD ON enables the synchronization word of the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD:SYMBOL

Sets or queries the synchronization word of the measurement source when [SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD is ON. The word depends on the modulation type selected by the [SENSe]:DDEMod[:SOURce<x>]:MODulation:TYPE command.

## Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD:SYMBOL<block>
```

```
[SENSe]:DDEMod[:SOURce<x>]:SYNCh:WORD:SYMBOL?
```

## Arguments

<x> = 1 to 8, representing Range of Source.

`<block> ::= #<num_digit><num_byte><sym(1)><sym(2)>...<sym(n)>`

Where

`<num_digit>` is the number of digits in `<num_byte>`.

`<num_byte>` is the number of bytes of data that follow.

`<sym(n)>` is the  $n^{\text{th}}$  symbol value of the sync word. 32-bit integer.

n: Max 256.

## Examples

`SENSe:DDEMOD:SOURce1:SYNCH:WORD:SYMBOL #216xxxx` (4 symbols) sets a sync word composed of four symbols for the measurement source1.

## [SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs

Sets or queries the fundamental unit of time of the measurement source.

### Conditions

Measurement views: General purpose digital modulation

Source keyword is optional, if not present Source 1 will be default source.

### Group

Sense commands

### Syntax

`[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs { SEConds | SYMBols }`

`[SENSe]:DDEMod[:SOURce<x>]:TIME:UNITs?`

### Arguments

`<x>` = 1 to 8, representing Range of Source.

`SEConds` specifies the fundamental unit of time as seconds.

`SYMBols` specifies the fundamental unit of time as symbols.

### Examples

`SENSe:DDEMOD:SOURce1:TIME:UNITs SEConds` specifies the fundamental unit of time as seconds for measurement source 1.

## [SENSe]:DIQVtime[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Demod I&Q versus Time for a selected Measurement view number.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).



## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Demod I&Q versus Time).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:DIQVtime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:DIQVtime[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

SENSe:DIQVtime:MEASview1:SELEcted:SOURce? queries the Source for Demod I&Q versus Time Measurement View 1.

## [SENSe]:DPX:AUDio:DEMod:GAIN

Sets or queries the audio gain.



**Note:** The sound level is also affected by the Windows volume control.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:AUDio:DEMod:GAIN <value>
```

```
[SENSe]:DPX:AUDio:DEMod:GAIN?
```

## Arguments

<value>::<NR1> specifies the audio gain. Range: 0 to 15 (integer).

## Examples

SENSe:DPX:AUDIO:DEMOD:GAIN 7 sets the audio gain to 7.

## **[SENSe]:DPX:AUDio:DEMod:RXBWidth**

Sets or queries the receiver bandwidth in the audio demodulation.

### **Conditions**

Measurement views: DPX, Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:DPX:AUDio:DEMod:RXBWidth <value>
```

```
[SENSe]:DPX:AUDio:DEMod:RXBWidth?
```

### **Arguments**

<value>::<Nrf> specifies the receiver bandwidth in the audio demodulation.

Range: 1 kHz to 500 kHz.

### **Examples**

```
SENSe:DPX:AUDIO:DEMOD:RXBWIDTH 30kHz sets the receiver bandwidth to 30kHz.
```

## **[SENSe]:DPX:AUDio:DEMod:RXFrequency? (Query Only)**

Returns the receiver frequency in the audio demodulation. The frequency depends on the setting of the [SENSe]:DPX:AUDio:DEMod:TUNE command.

### **Conditions**

Measurement views: DPX, Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:DPX:AUDio:DEMod:RXFrequency?
```

### **Arguments**

None

### **Returns**

<frequency>::<Nrf> is the receiver frequency in the audio demodulation.

## Examples

`SENSe:DPX:AUDIO:DEMOD:RXFREQUENCY?` might return `80.3E+6`, indicating that the receiver frequency is 80.3 MHz.

## [SENSe]:DPX:AUDIo:DEMod:STATe

Determines whether to enable the audio demodulation.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:AUDIo:DEMod:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:AUDIo:DEMod:STATe?
```

### Arguments

OFF or 0 disables the audio demodulation.

ON or 1 enables the audio demodulation.

## Examples

`SENSe:DPX:AUDIO:DEMOD:STATE ON` enables the audio demodulation.

## [SENSe]:DPX:AUDIo:DEMod:TUNE

Sets or queries how to determine the tuning frequency in the audio demodulation.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:AUDIo:DEMod:TUNE { MR | MARK1 | MARK2 | MARK3 | MARK4 | SMARker |  
FREQcontrol }
```

```
[SENSe]:DPX:AUDIo:DEMod:TUNE?
```

### Arguments

The following table lists the arguments.

**Table 50: Frequency tuning**

Argument	Tune with
MR	Reference marker (MR)
MARK1	Marker 1 (M1)
MARK2	Marker 2 (M2)
MARK3	Marker 3 (M3)
MARK4	Marker 4 (M4)
SMARker	Selected marker
FREQcontrol	Center frequency setting

**Examples**

`SENSe:DPX:AUDIO:DEMODO:TUNE MARK1` sets the tuning frequency to the value at Marker 1.

**[SENSe]:DPX:AUDIo:DEMod:TYPE**

Sets or queries the modulation type in the audio demodulation.

**Conditions**

Measurement views: DPX, Spectrum

**Group**

Sense commands

**Syntax**

`[SENSe]:DPX:AUDIo:DEMod:TYPE { AM | FM }`

`[SENSe]:DPX:AUDIo:DEMod:TYPE?`

**Arguments**

AM selects the AM (Amplitude Modulation).

FM selects the FM (Frequency Modulation).

**Examples**

`SENSe:DPX:AUDIO:DEMODO:TYPE FM` selects FM in the audio demodulation.

**[SENSe]:DPX:{BANDwidth|BWIDth}:ACTual? (Query Only)**

Queries the actual bandwidth whether set automatically or manually.

**Conditions**

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:{BANDwidth|BWIDth}:ACTual?
```

## Arguments

None

## Examples

`SENSe:DPX:BANDWIDTH:ACTUAL?` returns the RBW as a percentage of the Span.

## [SENSe]:DPX:{BANDwidth|BWIDth}:RESolution

Sets or queries the resolution bandwidth (RBW) in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:{BANDwidth|BWIDth}:RESolution <value>
```

```
[SENSe]:DPX:{BANDwidth|BWIDth}:RESolution?
```

## Arguments

`<value>::<NRf>` specifies the resolution bandwidth.

Range: 1 Hz to 10 MHz.

## Examples

`SENSe:DPX:BANDWIDTH:RESOLUTION?` might return `100.00003E+3` for the RBW.

## [SENSe]:DPX:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

## Arguments

OFF or 0 specifies that the resolution bandwidth is set manually using the [SENSe]:DPX:{BANDwidth|BWIDth}:ACTual? command.

ON or 1 specifies that the resolution bandwidth is set automatically.

## Examples

SENSe:DPX:BANDWIDTH:AUTO ON sets the resolution bandwidth automatically.

## [SENSe]:DPX:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:CLEar:RESults
```

## Arguments

None

## Examples

SENSe:DPX:CLEAR:RESULTS restarts multi-trace functions.

## [SENSe]:DPX:COLor

Sets or queries the color palette of three-dimensional graphs.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:COLor { RED | GREen | BLUe | CYAN | BCYan | YELLow | MAGenta | GRAY | TEMPerature | SPECtral }
```

```
[SENSe]:DPX:COLor?
```

## Arguments

The following table lists the arguments.

**Table 51: Color palette for DPX, Spectrum**

Argument	Palette
RED	Red
GREen	Green
BLUe	Blue
CYAN	Cyan
BCYan	Binary cyan
YELLow	Yellow
MAGenta	Magenta
GRAY	Gray
TEMPerature	Temperature
SPECtral	Spectral

## Examples

`SENSe:DPX:COLOR TEMPERATURE` selects the temperature color palette.

## [SENSe]:DPX:COLor:MAXimum

Sets or queries the maximum value of the color axis in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:COLor:MAXimum <value>
```

```
[SENSe]:DPX:COLor:MAXimum?
```

## Arguments

`<value> ::= <NRf>` specifies the maximum value of the color axis.

Range: The minimum value to 100%.

The minimum value is set using the `[SENSe]:DPX:COLor:MINimum` command.

## Examples

`SENSe:DPX:COLOR:MAXIMUM 90` sets the maximum value of the color axis to 90%.

## [SENSe]:DPX:COLor:MINimum

Sets or queries the minimum value of the color axis in the DPX, Spectrum measurement.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:COLor:MINimum <value>
```

```
[SENSe]:DPX:COLor:MINimum?
```

### Arguments

`<value>::<NRf>` specifies the minimum value of the color axis.

Range: 0% to the maximum value.

The maximum value is set using the `[SENSe]:DPX:COLor:MAXimum` command.

## Examples

`SENSe:DPX:COLOR:MINIMUM 10` sets the minimum value of the color axis to 10%.

## [SENSe]:DPX:DDENsity:MEASurement:SHOW

Sets the Show measurement value (measurement always visible for DPX Trigger) or queries for the current value.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:DDENsity:MEASurement:SHOW { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:DDENsity:MEASurement:SHOW?
```



## Arguments

OFF or 0 specifies that the Show measurement value is disabled.

ON or 1 specifies that the Show measurement value is disabled.

## Examples

`SENSe:DPX:DDENSITY:MEASUREMENT:SHOW ON` enables the Show measurement value.

## [SENSe]:DPX:DGRam:COLor:MAXimum

Sets or queries the maximum value of the color axis in the DPXogram measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:DGRam:COLor:MAXimum <value>`

`[SENSe]:DPX:DGRam:COLor:MAXimum?`

## Arguments

`<value> ::= <NRf>` specifies the maximum value of the color axis.

Range: -50 dBm to -169.00 dBm.

The minimum value is set using the `[SENSe]:DPX:DGRam:COLor:MINimum` command.

## Examples

`SENSe:DPX:DGRAM:COLOR:MAXIMUM 0` sets the maximum value of the color axis to 0.00 dBm.

## [SENSe]:DPX:DGRam:COLor:MINimum

Sets or queries the minimum value of the color axis in the DPXogram measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:DGRam:COLor:MINimum <value>`

[SENSe]:DPX:DGRam:COLor:MINimum?

### Arguments

<value> ::= <Nrf> specifies the maximum value of the color axis.

Range: 49.00 dBm to -170.00 dBm.

The maximum value is set using the [SENSe]:DPX:DGRam:COLor:MAXimum command.

### Examples

SENSe:DPX:DGRAM:COLOR:MINIMUM -100 sets the minimum value of the color axis to -100.00 dBm.

## [SENSe]:DPX:DGRam:POINTs:COUNT

Sets or queries the number of trace points acquired for the DPXogram display.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

[SENSe]:DPX:DGRam:POINTs:COUNT { P801 | P2401 | P4001 }

[SENSe]:DPX:DGRam:POINTs:COUNT?

### Arguments

P801 sets the number of sample points to 801.

P2401 sets the number of sample points to 2401.

P4001 sets the number of sample points to 4001.

### Examples

SENSe:DPX:DGRAM:POINTS:COUNT P2401 sets the number of trace points to acquire for the DPXogram display.

## [SENSe]:DPX:DGRam:TIME:CAPacity? (Query Only)

Returns the time capacity of the DPXogram measurement.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:TIME:CAPacity?
```

## Arguments

None

## Returns

<Time> ::= <String> The time capacity in hours:minutes:seconds.

## Examples

SENSe:DPX:DGRAM:TIME:CAPACITY? might return "0:25:50" indicating that the time capacity of the measurement was 0 hours, 25 minutes, and 50 seconds.

## [SENSe]:DPX:DGRam:TIME:RESolution

Sets or queries the time resolution of the DPXogram measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:TIME:RESolution <value>
```

```
[SENSe]:DPX:DGRam:TIME:RESolution?
```

## Arguments

<value> ::= <NRF> specifies the time resolution of the DPXogram measurement.

## Examples

SENSe:DPX:DGRAM:TIME:RESOLUTION 25.0 ms sets the time resolution to 25 ms.

## [SENSe]:DPX:DGRam:TIME:RESolution:AUTO

Determines whether to set the time resolution automatically or manually in the DPXogram measurement.

When Auto is enabled, the time resolution value is based on the Time/div setting. See the DISPLAY:DPX:DGRam:TIME[:SCALE]:PDIVision command.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:TIME:RESolution:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:DGRam:TIME:RESolution:AUTO?
```

## Arguments

OFF or 0 disables the auto time resolution.

ON or 1 enables the auto time resolution.

## Examples

SENSe:DPX:DGRAM:TIME:RESOLUTION:AUTO ON sets the timer resolution automatically.

## [SENSe]:DPX:DGRam:WATerfall:DIRection

Sets or queries the direction of the Waterfall (DPXogram) display.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:WATerfall:DIRection { NW | NE }
```

```
[SENSe]:DPX:DGRam:WATerfall:DIRection?
```

## Arguments

NW sets the display direction to northwest.

NE sets the display direction to northeast.

## Examples

SENSe:DPX:DGRAM:WATERFALL:DIRECTION NW sets the waterfall display direction to northwest.

## [SENSe]:DPX:DGRam:WATerfall:ENABLE

Sets or queries the Waterfall (DPXogram) display.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:WATERfall:ENABle { OFF | ON | 0 | 1 }
[SENSe]:DPX:DGRam:WATERfall:ENABle?
```

## Arguments

OFF or 0 disables the Waterfall display.

ON or 1 enables the Waterfall display.

## Examples

SENSe:DPX:DGRAM:WATERFALL:ENABLE ON enables the 3-D Waterfall display.

## [SENSe]:DPX:DGRam:Y[:SCALE]:AUTO (No Query Form)

Rescales the height axis automatically to fit the waveform to the screen in the DPXogram display.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:Y[:SCALE]:AUTO
```

## Arguments

None

## Examples

SENSe:DPX:DGRAM:Y[:SCALE]:AUTO rescales the height scale automatically to fit the waveform to the DPXogram display.

## [SENSe]:DPX:DGRam:Y[:SCALE]:RESet (No Query Form)

Resets the height scale of the DPXogram display to the default values:

Height position = 0 dB and Height scale = 100 dB.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DGRam:Y[:SCALE]:RESet
```

## Arguments

None

## Examples

`SENSe:DPX:DGRAM:Y[:SCALE]:RESET` Resets the height scale of the DPXogram display.

## [SENSe]:DPX:DWELI

Sets or queries the value of the Dwell time for the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DWELl <value>
```

```
[SENSe]:DPX:DWELl?
```

## Arguments

`<value>::<Nrf>` specifies the time the DPX sweep remains in a frequency segment collecting data and updating the bitmap and traces before moving on to the next higher frequency segment.

Range: 50 ms to 100 s per frequency segment.

## Examples

`SENSe:DPX:DWELl 100ms` sets the dwell value for sweeps to 100 ms.

## [SENSe]:DPX:DWELI:AUTO

Sets the value of the Dwell time automatically or queries for the current value.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:DWELl:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:DWELl:AUTO?
```

## Related Commands

```
[SENSe]:DPX:DWELl
```

## Arguments

OFF or 0 specifies that the dwell is set manually.

ON or 1 specifies that the dwell is set automatically.

## Examples

SENSe:DPX:DWELl:AUTO ON sets the dwell value automatically for sweeps.

## [SENSe]:DPX:FREQuency:CENTer

Sets or queries the center frequency in the DPX, Spectrum measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:FREQuency:CENTer <value>
```

```
[SENSe]:DPX:FREQuency:CENTer?
```

## Related Commands

```
[SENSe]:DPX:FREQuency:START, [SENSe]:DPX:FREQuency:STOP
```

## Arguments

<value> ::= <NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz

- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:DPX:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

## [SENSe]:DPX:FREQUENCY:SPAN

Sets or queries the frequency span in the DPX, Spectrum measurement.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

`[SENSe]:DPX:FREQUENCY:SPAN <value>`

`[SENSe]:DPX:FREQUENCY:SPAN?`

### Arguments

`<value> ::= <NRf>` is the frequency span.

Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

## Examples

`SENSe:DPX:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

## [SENSe]:DPX:FREQUENCY:SPAN:{BANDwidth|BWIDth}[:RESolution]:RATio

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the DPX, Spectrum measurement.

### Conditions

Measurement views: DPX, Spectrum

### Group

Sense commands

### Syntax

`[SENSe]:DPX:FREQUENCY:SPAN:{BANDwidth|BWIDth}[:RESolution]:RATio <value>`

`[SENSe]:DPX:FREQUENCY:SPAN:{BANDwidth|BWIDth}[:RESolution]:RATio?`



## Related Commands

[SENSe]:DPX:{BANDwidth|BWIDth}[:RESolution]:AUTO

## Arguments

<value> ::= <NRf> is used to calculate the RBW.

## Examples

SENSe:DPX:FREQuency:SPAN:BANDWIDTH:RATIO 100 the ratio of 100 is used to set the resolution bandwidth when  
SENSe:DPX:{BANDwidth|BWIDth}[:RESolution]:AUTO is set to ON.

## [SENSe]:DPX:FREQuency:START

Sets or queries the measurement start frequency (left edge on the graph) in the DPX, Spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:DPX:FREQuency:CENTer command.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

[SENSe]:DPX:FREQuency:START <value>

[SENSe]:DPX:FREQuency:START?

## Related Commands

[SENSe]:DPX:FREQuency:STOP

## Arguments

<value> ::= <NRf> is the measurement start frequency.

Range: (center frequency) ± (span)/2.

## Examples

SENSe:DPX:FREQUENCY:START 6.95GHz sets the start frequency to 6.95 GHz.

## [SENSe]:DPX:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:DPX:FREQuency:STEP:AUTO OFF.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:FREQuency:STEP <value>
```

```
[SENSe]:DPX:FREQuency:STEP?
```

## Related Commands

```
[SENSe]:DPX:FREQuency:STEP:AUTO
```

## Arguments

<value> ::= <NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:DPX:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

## [SENSe]:DPX:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:DPX:FREQuency:STEP:AUTO?
```

## Arguments

OFF or 0 specifies that the frequency step size is set manually using the `[SENSe]:DPX:FREQuency:STEP` command.

ON or 1 specifies that the frequency step size is set automatically.

## Examples

`SENSE:DPX:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

## [SENSe]:DPX:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the DPX, Spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:DPX:FREQuency:CENTer` command.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:FREQuency:STOP <value>`

`[SENSe]:DPX:FREQuency:STOP?`

## Related Commands

`[SENSe]:DPX:FREQuency:START`

## Arguments

`<value>::<NRf>` is the measurement stop frequency.

Range:  $(\text{center frequency}) \pm (\text{span})/2$ .

## Examples

`SENSe:DPX:FREQUENCY:STOP 7.05GHz` sets the stop frequency to 7.05 GHz.

## [SENSe]:DPX:PLOT

Sets or queries the plot in the DPX, Spectrum measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:PLOT { DGRam | SPLit | SPECTrum | ZSPan | FREQuency | PHASe }`

`[SENSe]:DPX:PLOT?`

## Arguments

DGRam selects the DPXogram display.

SPLit selects the DPX Split display.

SPECtrum selects the DPX, Spectrum display.

ZSPan select the DPX Zero Span display.

FREquency selects the DPX Frequency vs Time display.

PHASe selects the DPX Phase vs Time display.

## Examples

SENSe:DPX:PLOT SPECTRUM selects the DPX, Spectrum display

## [SENSe]:DPX:POINTs:COUNT

Sets the number of trace points acquired for the DPX, Spectrum display.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:POINTs:COUNT { P801 | P2401 | P4001 | P10401 }
```

```
[SENSe]:DPX:POINTs:COUNT?
```

## Arguments

P801 sets the number of sample points to 801.

P2401 sets the number of sample points to 2401.

P4001 sets the number of sample points to 4001.

P8001 sets the number of sample points to 8001.

P10401 sets the number of sample points to 10401.

## Examples

SENSe:DPX:POINTS:COUNT P2401 sets the number of trace points to acquire for the DPX, Spectrum display.

## [SENSe]:DPX:TDM:{BANDwidth|BWIDth}

Sets or queries the value of the measurement bandwidth in the DPX measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSE]:DPX:TDM:{BANDwidth|BWIDth} <value>
```

```
[SENSE]:DPX:TDM:{BANDwidth|BWIDth}?
```

## Arguments

<value>::<NRf> specifies the measurement bandwidth.

Range: 100 Hz to Max AcqBW.

## Examples

`SENSe:DPX:TDM:BANDWIDTH 200` sets the measurement bandwidth to 200 Hz.

## [SENSe]:DPX:TDM:{BANDwidth|BWIDth}:ACTual? (Query Only)

Queries the actual bandwidth in the DPX measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSE]:DPX:TDM:{BANDwidth|BWIDth}:ACTual?
```

## Arguments

None

## Returns

<NRf> The actual RBW in Hz.

## Examples

`SENSe:DPX:TDM:BANDWIDTH:ACTUAL?` might return 10E+3, indicating that the actual bandwidth is 10 kHz

## [SENSe]:DPX:TDM:FREQuency:CENTer

Sets or queries the center frequency in the DPX, Spectrum measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSE]:DPX:TDM:FREQUENCY:CENTER <value>
```

```
[SENSe]:DPX:TDM:FREQUENCY:CENTER?
```

## Arguments

<value>::<Nrf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:DPX:TDM:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

## [SENSe]:DPX:TDM:FREQUENCY:TXBer:CLEAr:RESults (No Query Form)

Clears the measurement results of the active Tx BER measurement. Sending this command does not halt the Tx BER measurement.

## Conditions

Measurement views: DPX, frequency

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:TDM:FREQUENCY:TXBer:CLEAr:RESults
```

## Arguments

None

## Examples

`SENSe:DPX:TDM:FREQUENCY:TXBER:CLEAR:RESULTS` clears the results of the active TxBER measurement, but does not halt the measurement.

## [SENSe]:DPX:TDM:FREQuency:TXBer:ENABle

Sets or queries the Tx BER measurement.

### Conditions

Measurement views: DPX, frequency

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:TDM:FREQuency:TXBer:ENABle { OFF | ON | 0 | 1 }
```

### Arguments

OFF or 0 halts the Tx BER measurement.

ON or 1 starts the Tx BER measurement.

### Returns

## Examples

```
SENSe:DPX:TDM:FREQUENCY:TXBER:ENABLE
```

## [SENSe]:DPX:TDM:FREQuency:TXBer:PATTern

Sets or queries the Tx BER data pattern.

### Conditions

Measurement views: DPX frequency

### Group

Sense commands

### Syntax

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATTern { PRBS9 | PRBS11 | PRBS15 | USER }
```

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATTern?
```

### Arguments

PRBS9 specifies a 9-bit PRBS as the data pattern.

PRBS11 specifies a 11-bit PRBS as the data pattern.

PRBS15 specifies a 15-bit PRBS as the data pattern

USER specifies a user-defined sequence as the data pattern.

## Returns

<string> which is the selected data pattern. Returned strings are the same as the arguments.

## Examples

SENSe:DPX:TDM:FREQUENCY:TXBER:PATTERN PRBS9 sets the data pattern to PRBS9.

## [SENSe]:DPX:TDM:FREQUENCY:TXBER:PATTERN:INVERT

Sets or queries the data pattern Invert setting.

## Conditions

Measurement views: DPX frequency

## Group

Sense commands

## Syntax

[SENSe]:DPX:TDM:FREQUENCY:TXBER:PATTERN:INVERT { 0 | 1 | OFF | ON }

[SENSe]:DPX:TDM:FREQUENCY:TXBER:PATTERN:INVERT?

## Arguments

OFF or 0 sets the data pattern to normal.

ON or 1 inverts the data pattern.

## Returns

<value> ::= <NR1> where:

0 indicates the data pattern is normal.

1 indicates the data pattern is inverted.

## Examples

SENSe:DPX:TDM:FREQUENCY:TXBER:PATTERN:INVERT 1 inverts the data pattern.

## [SENSe]:DPX:TDM:FREQUENCY:TXBER:PATTERN:REPEAT

Sets or queries the test duration for a User Pattern. A User Pattern test duration can be one time through or it can repeat.



## Conditions

Measurement views: DPX frequency

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REPeat { 0 | 1 | OFF | ON }
```

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REPeat?
```

## Arguments

OFF or 0 sets the data pattern to normal.

ON or 1 inverts the data pattern.

## Returns

<value> ::= <NR1> where:

0 indicates repeat is off.

1 indicates repeat is on.

## Examples

SENSe:DPX:TDM:FREQUENCY:TXBER:PATTERN:REPEAT 1 causes the User pattern to repeat.

## [SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REVerse

Sets or queries whether the data pattern is in normal order or reversed. This setting applies only to PRBS patterns.

## Conditions

Measurement views: DPX frequency

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REVerse { 0 | 1 | OFF | ON }
```

```
[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:REVerse?
```

## Arguments

OFF or 0 sets the data pattern to normal order.

ON or 1 sets the data pattern to reversed order.

## Returns

<value> ::= <NR1> where:

0 indicates the data pattern is in normal order.

1 indicates the data pattern is in reversed order.

## Examples

SENSe:DPX:TDM:FREQUENCY:TXBER:PATTERN:REVERSE? might return 1, indicating the data pattern is reversed.

## [SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:SOURce? (Query Only)

Returns the path and file name of the user-defined pattern file.

## Conditions

Measurement views: DPX frequency

## Group

Sense commands

## Syntax

[SENSe]:DPX:TDM:FREQuency:TXBer:PATtern:SOURce?

## Returns

<path>::<string> which is the path to the defined User data pattern.

## Examples

SENSe:DPX:TDM:FREQUENCY:TXBER:PATTERN:SOURCE? might return "C:\RSA5100B Files\Example Files\SampleUserPattern.csv".

## [SENSe]:DPX:TDM:RBW

Sets or queries the value of the RBW filter.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

[SENSe]:DPX:TDM:RBW <value>

[SENSe]:DPX:TDM:RBW?

## Arguments

<value> : : <NRf> specifies the RBW filter value.

Range: 1 Hz to 60MHz

## Examples

`SENSe:DPX:TDM:RBW 2.5` sets the RBW filter value to 2.5 Hz.

## [SENSe]:DPX:TDM:RBW:STATe

Determines whether to set the resolution bandwidth (RBW) filter automatically or manually in the DPX measurement.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:TDM:RBW:STATe { OFF | ON | 0 | 1 }`

`[SENSe]:DPX:TDM:RBW:STATe?`

## Arguments

OFF or 0 specifies that the RBW is set automatically.

ON or 1 specifies that the RBW is set manually.

## Examples

`SENSe:DPX:TDM:RBW:STATE ON` sets the RBW manually.

## [SENSe]:DPX:TDM:SWEep:TIME

Sets or queries the sweep time when the plot is set to ZSPan, FREQuency, or PHASe.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:DPX:TDM:SWEep:TIME <value>`

`[SENSe]:DPX:TDM:SWEep:TIME?`

## Arguments

<value> : : <NRf> specifies the sweep time.

Range: 100 ns to 2000 s.

## Examples

`SENSe:DPX:TDM:SWEEP:TIME 200` sets the sweep time to 200 s.

## [SENSe]:DPX:TDM:TMOTion

Sets or queries the trace motion when the plot is set to ZSPan, FREQuency, or PHASe.

## Conditions

Measurement views: DPX, Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:DPX:TDM:TMOTion { NONE | ROLL | NORMal }
```

```
[SENSe]:DPX:TDM:TMOTion?
```

## Arguments

`NONE` turns scrolling off.

`ROLL` shifts the previous data points to the left and the latest data point to the right edge of the display.

`NORMAL` shifts the latest data point on the display from left to right.

## Examples

`SENSe:DPX:TDM:TMOTION NORMAL` sets the scroll direction to normal.

## [SENSe]:EDiagram[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Eye diagram for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Eye diagram).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:EDIagram[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:EDIagram[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

SENSe:EDIagram:MEASview1:SELEcted:SOURce?, queries the Source for Eye diagram Measurement View 1.

## [SENSe]:DEMOd:EVM[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of EVM versus Time for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Eye diagram).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:DEMOd:EVM[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:DEMOd:EVM[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

SENSe:DEMOd:EVM:MEASview1:SELEcted:SOURce? queries the Source for EVM versus Time Measurement View 1.

## **SENSe:EMC:ACCEssory<x>:ANTEнна:DISTance**

Sets or queries the distance between EUT and Antenna in EMC View.

<x> = <NR1> 1 to 6

### **Conditions**

Measurement views: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:ACCEssory<x>:ANTEнна:DISTance <value>
```

```
SENSe:EMC:ACCEssory<x>:ANTEнна:DISTance?
```

### **Arguments**

<value> = <NR3>

### **Returns**

<NR3>

### **Examples**

`SENSe:EMC:ACCEssory1:ANTEнна:DISTance` might returns 10.5, indicating that the distance between accessory 1 (Antenna) and EUT is 10.5 m.

## **SENSe:EMC:ACCEssory<x>:ASSEt:TAG**

Sets or queries the asset tag of the specified accessory in EMC View.

<x> = <NR1> 1 to 6

### **Conditions**

Measurement views: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:ACCEssory<x>:ASSEt:TAG <arg>
```

```
SENSe:EMC:ACCEssory<x>:ASSEt:TAG?
```

### **Arguments**

<arg> = "String"

## Returns

<String>

## Examples

`SENSe:EMC:ACCEssory1:ASSEt:TAG` might return A4001993, indicating that asset tag of accessory 1 is A4001993.

## SENSe:EMC:ACCEssory<x>:CALibration:DATE

Sets or queries the Calibration date of the specified accessory in EMC View.

<x> = <NR1> 1 to 6

## Conditions

Measurement views: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:ACCEssory<x>:CALibration:DATE <arg>`

`SENSe:EMC:ACCEssory<x>:CALibration:DATE?`

## Arguments

<arg> = "String"

## Returns

<String>

## Examples

`SENSe:EMC:ACCEssory1:CALibration:DATE` might return "9/21/2017", indicating that calibration date of the accessory 1 is 9/21/2017 (21st of September 2017).

## SENSe:EMC:ACCEssory<x>:CALibration:DATE:DISAbLe

Sets or queries the enabling and disabling the calibration due date of the specified accessory in EMC View.

<x> = <NR1> 1 to 6

## Conditions

Measurement views: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:ACCEssory<x>:CALibration:DATE:DISAble { 0 | OFF | 1 | ON}
```

```
SENSe:EMC:ACCEssory<x>:CALibration:DATE:DISAble?
```

## Arguments

0 or OFF disables the calibration due date check box

1 or ON enables the calibration due date check box

## Returns

<Boolean>0,1

## Examples

SENSe:EMC:ACCEssory1:CALibration:DATE:DISAble 0 will disable the calibration due date check box of accessory 1.

## SENSe:EMC:ACCEssory<x>:CONStant:LOSS

Sets or queries the Constant loss of an accessory in EMC View.

<x> = <NR1> 1 to 6

## Conditions

Measurement views: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:ACCEssory<x>:CONStant:LOSS <value>
```

```
SENSe:EMC:ACCEssory<x>:CONStant:LOSS?
```

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

SENSe:EMC:ACCEssory1:CONStant:LOSS might return 10.00, indicating that the Constant Loss of accessory 1 is 10 dB.

## SENSe:EMC:ACCEssory<x>:ENABLE

Sets or queries enabling and disabling the specified accessory in EMC View.



<x> = <NR1> 1 to 6

### Conditions

Measurement views: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:ACCEssory<x>:ENABle { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:ACCEssory<x>:ENABle?
```

### Arguments

0 or OFF disables the accessory

1 or ON enables the accessory

### Returns

<Boolean>0,1

### Examples

SENSe:EMC:ACCEssory<x>:ENABle 0 will disable accessory 1.

## SENSe:EMC:ACCEssory<x>:FREQuency:INTerpolation

Sets or queries the frequency interpolation of an accessory in EMC View.

<x> = <NR1> 1 to 6

### Conditions

Measurement views: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:ACCEssory<x>:FREQuency:INTerpolation { LIN | LOG }
```

```
SENSe:EMC:ACCEssory<x>:FREQuency:INTerpolation?
```

### Arguments

LIN = Sets Linear as frequency interpolation

LOG = Sets Log as frequency interpolation

## Returns

<String> LIN or LOG

## Examples

`SENSe:EMC:ACCEssory1:ANTEнна:DIStance` might returns 10.5, indicating that the distance between accessory 1 (Antenna) and EUT is 10.5 m.

## SENSe:EMC:ACCEssory<x>:LOSS:TABLE:ADD (No Query Form)

Adds a row in the loss table with values for the specified accessory in EMC view.

<x> = <NR1> 1 to 6



**Note:** This command works only when the specified accessory is enabled.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:ACCEssory<x>:LOSS:TABLE:ADD <arg1, arg2>`

## Arguments

<arg1> = Specifies the frequency value

<arg2> = Specifies the Gain/Loss value

## Returns

None

## Examples

`SENSe:EMC:ACCEssory1:LOSS:TABLE:ADD 30KHz,10` adds a new row to the accessory 1 loss table with frequency 30 kHz and an RBW of 10 k.

## SENSe:EMC:ACCEssory<x>:LOSS:TABLE:DELeTe:ALL (No Query Form)

Deletes all the row in the loss table for the specified accessory in EMC view.

<x> = <NR1> 1 to 6

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:ACCEssory<x>:LOSS:TABLE:DELeTe:ALL
```

## Arguments

None

## Returns

None

## Examples

`SENSe:EMC:ACCEssory2:LOSS:TABLE:DELeTe:ALL` deletes all the rows in the loss table of accessory 2.

## SENSe:EMC:ACCEssory<x>:LOSS:TYPE

Sets or queries the loss type of an accessory in EMC View.

<x> = <NR1> 1 to 6

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:ACCEssory<x>:LOSS:TYPE {ALT | CLOS | AFT}
```

```
SENSe:EMC:ACCEssory<x>:LOSS:TYPE?
```

## Arguments

ALT = Antenna Loss Table

CLOS = Constant Loss

AFT = Antenna Factor

## Returns

<String> ALT or CLOS or AFT

## Examples

`SENSe:EMC:ACCEssory1:LOSS:TYPE` might return CLOS, indicating that the loss type of accessory 1 is CLOS (Constant Loss).

## **SENSe:EMC:ACCEssory<x>:MODEl**

Sets or queries the model number of the specified accessory in EMC View.

<x> = <NR1> 1 to 6

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:ACCEssory<x>:MODEl <arg>
```

```
SENSe:EMC:ACCEssory<x>:MODEl?
```

### **Arguments**

<arg> = "String"

### **Returns**

<String>

### **Examples**

SENSe:EMC:ACCEssory2:MODEl might return A16235, indicating that the model number of accessory 2 is A16235.

## **SENSe:EMC:ACCEssory<x>:SERIAL**

Sets or queries the serial number of the specified accessory in EMC View.

<x> = <NR1> 1 to 6

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:ACCEssory<x>:SERIAL <arg>
```

```
SENSe:EMC:ACCEssory<x>:SERIAL?
```

### **Arguments**

<arg> = "String"

## Returns

<String>

## Examples

`SENSe:EMC:ACCEssory1:SERial` might return S1987652, indicating that the serial number of accessory 1 is S1987652.

## SENSe:EMC:ACCEssory<x>:TYPE

Sets or queries the type of specified accessory in EMC view.

<x> = <NR1> 1 to 6

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:ACCEssory<x>:TYPE <arg>`

`SENSe:EMC:ACCEssory<x>:TYPE?`

## Arguments

If emission type is Conducted:

If emission type is Radiated:

<arg> = { ANTE | NFPR | CABL | AMPL | LIMI | ATTE | FILT | OTHE }

where

LISN = LISN

CABL = Cable

LIMI = Limiter

AMPL = Amplifier

ATTE = Attenuator

FILT = Filter

OTHE = Others

ANTE = Antenna

NFPR = Near Field Probe

## Returns

<String> See Arguments

## Examples

`SENSe:EMC:ACCEssory1:TYPE` might return `LIMI`, indicating that accessory 1 is Limiter.

## SENSe:EMC:CLEAr:RESults (No Query Form)

Clears all the data in EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:CLEAr:RESults
```

### Arguments

None

### Returns

None

## Examples

`SENSe:EMC:CLEAr:RESults` clears the data in the EMC-EMI display.

## SENSe:EMC:EMISsion:TYPE

Sets or queries the type of emission in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:EMISsion:TYPE {COND | RADI}
```

```
SENSe:EMC:EMISsion:TYPE?
```

### Arguments

`COND` = Sets Conducted as Emission type

`RADI` = Sets Radiated as Emission type

## Returns

<String> COND or RAD1

## Examples

`SENSE:EMC:EMISSION:TYPE` sets Conducted as the emission type.

## SENSe:EMC[:FREQUENCY]:OVERlap? (Query Only)

Queries whether there is a frequency range overlap in ranges and Limits of EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC[:FREQUENCY]:OVERlap?`

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC[:FREQUENCY]:OVERLAP?` might return 1, indicating that the frequency is overlapped.

## SENSe:EMC:LIST

Sets or queries the list of spots to be displayed in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:LIST {ALL | OVER | OLWM}`

`SENSe:EMC:LIST?`

## Arguments

ALL = Sets All spots to be displayed

OVER = Sets Over Limit spots to be displayed

OLWM = Sets Over Limit with margin spots to be displayed

## Returns

<String> See Arguments

## Examples

SENSE:EMC:LIST might return ALL, indicating All spots are displayed.

## SENSe:EMC:MARGin

Sets or queries the Margin value in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

SENSe:EMC:MARGin <value>

SENSe:EMC:MARGin?

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

SENSE:EMC:MARGIN might return 15, indicating that the margin value is 15 dB.

## SENSe:EMC:MEASurement:AMBIent:DELTA:THREShold

Sets or queries Delta threshold value of Ambient detection in EMC View.

## Conditions

Measurement view: EMC - EMI Display



## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:AMBIent:DELTA:THREshold <value>
```

```
SENSe:EMC:MEASurement:AMBIent:DELTA:THREshold?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

SENSe:EMC:MEASUREMENT:AMBIENT:DELTA:THRESHOLD might return 1.00000, indicating that the Delta threshold value of ambient detection is 1.0 dB.

## SENSe:EMC:MEASurement:AMBIent:DETEctor

Sets or queries the detector type of the ambient detector in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:AMBIent:DETEctor <arg>
```

```
SENSe:EMC:MEASurement:AMBIent:DETEctor?
```

## Arguments

<arg> =

AVER = Sets Avg (VRMS) as the detector type

PEAK = Sets +Peak as the detector type

MILSTD = Sets Mil +Peak as the detector type

QUAS = Sets CISPR QPk as the detector type

Sets CISPR +Pk as the detector type

CAV = Sets CISPR Avg as the detector type

AVGL = Sets Avg (of logs) as the detector type

CAVL = Sets CISPR Avg (of logs) as the detector type

DSA = Sets DEF STAN Avg as the detector type

DSP = Sets DEF STAN Peak as the detector type

### Returns

<String> See Argument

### Examples

SENSE:EMC:MEASUREMENT:AMBIENT:DETECTOR might return DSP, indicating that the ambient detector type is DEF STAN Peak.

## SENSe:EMC:MEASurement:AMBIent:DWELI:TIME

Sets or queries the dwell time of ambient detection in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

SENSe:EMC:MEASurement:AMBIent:DWELI:TIME <value>

SENSe:EMC:MEASurement:AMBIent:DWELI:TIME?

### Arguments

<value> = <NRf>

### Returns

<NRf>

### Examples

SENSE:EMC:MEASUREMENT:AMBIENT:DWELL:TIME might return 329.142E-6, indicating that the dwell time of ambient detection is 329.1  $\mu$ s.

## SENSe:EMC:MEASurement:AMBIent:DWELI:TIME:AUTO

Sets or queries to automatically set the dwell time of ambient detection in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:AMBIent:DWELL:TIME:AUTO { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:AMBIent:DWELL:TIME:AUTO?
```

## Arguments

0 or OFF disables the Auto check box of dwell time

1 or ON enables the Auto check box of dwell time

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASUREMENT:AMBIENT:DWELL:TIME:AUTO might return 1, indicating that the dwell time of ambient detection is set automatically.

## SENSe:EMC:MEASurement:AMBIent:START (No Query Form)

Starts the ambient detection with respect to selected detector.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:AMBIent:START
```

## Arguments

None

## Returns

None

## Examples

SENSe:EMC:MEASUREMENT:AMBIENT:START SINGLE performs a single ambient scan with respect to the selected scan detector.

## SENSe:EMC:MEASurement:INSpect:DETEctor<x>

Sets or queries the detector type for the specified detector in discrete search of EMC View.

<x> = 1 or 2 or 3 specifies detector number

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x> <arg>
```

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x>?
```

## Arguments

<arg> =

AVER = Sets Avg (VRMS) as the detector type

PEAK = Sets +Peak as the detector type

MILSTD = Sets Mil +Peak as the detector type

QUAS = Sets CISPR QPk as the detector type

CPE = Sets CISPR +Pk as the detector type

CAV = Sets CISPR Avg as the detector type

AVGL = Sets Avg (of logs) as the detector type

CAVL = Sets CISPR Avg (of logs) as the detector type

DSA = Sets DEF STAN Avg as the detector type

DSP = Sets DEF STAN Peak as the detector type

## Returns

<String> See Argument

## Examples

SENSe:EMC:MEASurement:INSpect:DETEctor1 might return QUAS, indicating that the detector type for detector 1 in discrete search is CISPR QPk.

## SENSe:EMC:MEASurement:INSpect:DETEctor<x>:COMParision:LEVEL

Sets or queries the comparison level value for the specified detector in discrete search of EMC View.

<x> = 1 or 2 or 3 (specifies detector number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x>:COMParision:LEVEL <value>
```

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x>:COMParision:LEVEL?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

SENSe:EMC:MEASurement:INSpect:DETEctor1:COMParision:LEVEL might return 39.9897, indicating that the comparison level value of detector 1 is 39.98 dBuV.

## SENSe:EMC:MEASurement:INSpect:DETEctor<x>:ENABLE

Sets or queries the Enabling or disabling of specified detector in discrete search of EMC View.

<x> = 1 or 2 or 3 (specifies detector number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x>:ENABLE { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:INSpect:DETEctor<x>:ENABLE?
```

## Arguments

0 or OFF disables the specified detector

1 or ON enables the specified detector

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASurement:INSpect:DETEctor1:ENABLE might return 1, indicating detector 1 is enabled.

## SENSe:EMC:MEASurement:INSpect:DWELL:TIME

Sets or queries the dwell time in discrete search of EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:DWELL:TIME <value>
```

```
SENSe:EMC:MEASurement:INSpect:DWELL:TIME?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

SENSe:EMC:MEASUREMENT:INSPECT:DWELL:TIME might return 2.500, indicating that the dwell time is 2.5 s.

## SENSe:EMC:MEASurement:INSpect:DWELL:TIME:AUTO

Sets or queries to automatically set the dwell time in discrete search of EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:DWELL:TIME:AUTO { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:INSpect:DWELL:TIME:AUTO?
```

## Arguments

0 or OFF disables the auto check box

1 or ON enables the auto check box

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASUREMENT:INSPECT:DWELL:TIME:AUTO might return 0, indicating auto check box is disabled.

## SENSe:EMC:MEASurement:INSpect:MODE

Sets or queries the inspect search mode in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE { DISC | HARM }
```

```
SENSe:EMC:MEASurement:INSpect:MODE?
```

### Arguments

DISC = Sets inspect search to Discrete mode

HARM = Sets inspect search to Harmonic mode

### Returns

<String> DISC or HARM

### Examples

SENSe:EMC:MEASUREMENT:INSPECT:MODE might return HARM, indicating inspect search is in harmonic search mode.

## SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:ENABLE

Sets or queries the enabling or disabling of the specified row number in discrete search of EMC View.

<x> = <NR1> specifies row number

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:ENABLE { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:ENABLE?
```

### Arguments

0 or OFF disables the specified row number

1 or ON enables the specified row number

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency3:ENABle?` might return 1, indicating that the row 3 is enabled.

## **SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency<x>:FREQuency**

Sets or queries the frequency value of the specified row number in discrete search of EMC View.

<x> = <NR1> specifies row number

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency<x>:FREQuency <value>`

`SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency<x>:FREQuency?`

## Arguments

<value> = <NRf> frequency value

## Returns

<NRf>

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency2:FREQuency` might return 25.00000E+6, indicating that the frequency value of row 2 is 25.00 MHz.

## **SENSe:EMC:MEASurement:INSpect:MODE:DIScrete:FREQuency<x>:RBW**

Sets or queries the RBW value of the specified row number in discrete search of EMC View.

<x> = <NR1> Specifies row number

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands



## Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:RBW <value>
```

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:RBW?
```

## Arguments

<value> = <NRf> RBW value

## Returns

<NRf>

## Examples

SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency1:RBW? might return 9.00000E+3, indicating that the RBW value of row 1 is 9.00 kHz.

## SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:RBW:AUTO

Sets or queries the enabling or disabling of Auto RBW check box of the specified row number in discrete search of EMC View.

<x> = <NR1> specifies row number

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:RBW:AUTO { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency<x>:RBW:AUTO?
```

## Arguments

0 or OFF disables the Auto check box

1 or ON enables the Auto check box

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:FREQuency4:RBW:AUTO? might return 1, indicating that the Auto check box for row 4 is enabled.

## **SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:LOAD:EMISsion:RESults (No Query Form)**

Loads the frequency spots from Emission results into discrete search table. Only the top 20 spots of emission results will be loaded.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:MEASurement:INSpect:MODE:DISCcrete:LOAD:EMISsion:RESults
```

### **Arguments**

None

### **Returns**

None

### **Examples**

SENSe:EMC:MEASUREMENT:INSPECT:MODE:DISCRETE:LOAD:EMISSION:RESULTS loads the emission results.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:ENABLE**

Sets or queries to enable or disable the specified harmonic in EMC View.

<x> = 1 or 2 or 3 specifies the harmonic

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:ENABLE { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:ENABLE?
```

### **Arguments**

0 or OFF disables the specified harmonic

1 or ON enables the specified harmonic

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:ENABle?` might return 1, indicating that the harmonic 1 is enabled.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:ENABle**

Sets or queries the enabling or disabling harmonic factor of a specified harmonic in EMC View.

<x> = 1 or 2 or 3 specifies the harmonic

<y> = <NR1> (1 to 10) Specifies the harmonic factor

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:ENABle { 0 | OFF | 1 | ON }`

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:ENABle?`

## Arguments

0 or OFF disables the harmonic factor

1 or ON enables the harmonic factor

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:FACTor2:ENABle` might return 1, indicating that harmonic factor 2 of harmonic 1 is enabled.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:FREQuency**

Sets or queries the frequency value of the specified harmonic and factor in harmonic search of EMC View.

p<x> = 1 or 2 or 3 specifies the harmonic

<y> = <NR1> (1 to 10) specifies the harmonic factor

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:FREQuency <value>
```

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:FREQuency?
```

## Arguments

<value> = <NRf> Frequency value

## Returns

<NRf>

## Examples

SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:FACTor2:FREQuency? might return 25.00000E+6, indicating that the frequency value of harmonic 1 factor 2 is 25.00 MHz.

## SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:HARMonic:FACTor

Sets or queries the harmonic factor value for the specified harmonic in EMC View.

<x> = 1 or 2 or 3 specifies the harmonic

<y> = <NR1> (1 to 10) Specifies the harmonic factor

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:HARMonic:FACTor <value>
```

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:HARMonic:FACTor?
```

## Arguments

<value> =

Either

OEIG = 1/8

QUAR = 1/4

HALF = 1/2

ONE = 1

TWO = 2

THREE = 3

FOUR = 4

FIVE = 5

SIX = 6

SEVEN = 7

EIGHT = 8

NINE = 9

TEN = 10

## Returns

<String> See Arguments

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:FACTor2:HARMonic:FACTor` might return "HALF", indicating the 2nd harmonic factor value for harmonic 1 is ½.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW**

Sets or queries the RBW value of the specified harmonic and factor in harmonic search of EMC View.

<x> = 1 or 2 or 3 specifies the harmonic

<y> = <NR1> (1 to 10) specifies the harmonic factor

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW<value>`

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW?`

## Arguments

<value> = <NRf> RBW value

## Returns

<NRf>

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:FACTor2:RBW?` might return `9.00000E+3`, indicating that the RBW value of harmonic 1 factor 2 is 9.00 kHz.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW:AUTO**

Sets or queries the enabling and disabling of Auto RBW check box with respect to the specified harmonic and factor in harmonic search of EMC View.

<x> = 1 or 2 or 3 specifies the harmonic

<y> = <NR1> (1 to 10) specifies the harmonic factor

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW:AUTO { 0 | OFF | 1 | ON }`

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FACTor<y>:RBW:AUTO?`

## Arguments

0 or OFF disables the Auto checkbox

1 or ON enables the Auto checkbox

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:MEASurement:INSpect:MODE:HARMonic2:FACTor1:RBW:AUTO?` might return 1, indicating that the Auto check box is enabled for harmonic 2 factor 1.

## **SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FREquency**

Sets or queries the frequency of the Specified Harmonic in EMC View.

<x> = 1 or 2 or 3 Specifies the harmonic

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FREquency <value>
```

```
SENSe:EMC:MEASurement:INSpect:MODE:HARMonic<x>:FREquency?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

SENSe:EMC:MEASurement:INSpect:MODE:HARMonic1:FREquency returns 20000, indicating that the frequency of harmonic 1 is 20 kHz.

## SENSe:EMC:MEASurement:INSpect:START (No Query Form)

Sets the type of run in inspect section of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:INSpect:START {SINGle | CONTinuous | STOP}
```

## Arguments

SINGle = Performs Single run

CONTinuous = Performs a Continuous run

STOP = Stops the acquisition

## Returns

None

## Examples

SENSe:EMC:MEASUREMENT:INSPECT:START SINGle sets the inspect mode to perform a single run.

## **SENSe:EMC:MEASurement:SCAN:DETEctor<x>**

Sets or queries the detector type of the specified scan detector in EMC View.

<x> = 1 or 2

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

SENSe:EMC:MEASurement:SCAN:DETEctor<x> <arg>

SENSe:EMC:MEASurement:SCAN:DETEctor<x>?

### **Arguments**

<arg> =

AVER = Sets Avg (VRMS) as the detector type

PEAK = Sets +Peak as the detector type

MILSTD = Sets Mil +Peak as the detector type

QUAS = Sets CISPR QPk as the detector type

CPE = Sets CISPR +Pk as the detector type

CAV = Sets CISPR Avg as the detector type

AVGL = Sets Avg (of logs) as the detector type

CAVL = Sets CISPR Avg (of logs) as the detector type

DSA = Sets DEF STAN Avg as the detector type

DSP = Sets DEF STAN Peak as the detector type

### **Returns**

<String> See Arguments

### **Examples**

SENSe:EMC:MEASurement:SCAN:DETEctor1? might return QUAS, indicating that the detector type for scan detector 1 is CISPR QPk.

## **SENSe:EMC:MEASurement:SCAN:DETEctor<x>:ENABle**

Sets or queries the enabling or disabling of the specified scan detector in EMC View.

<x> = 1 or 2



## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SCAN:DETEctor<x>:ENABle { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:SCAN:DETEctor<x>:ENABle?
```

## Arguments

0 or OFF disables the scan detector

1 or ON enables the scan detector

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASurement:SCAN:DETEctor1:ENABle? might return 1, indicating that scan detector 1 is enabled.

## SENSe:EMC:MEASurement:SCAN:DETEctor<x>:LIMit

Sets or queries the limit of specified scan detector in EMC View.

<x> = 1 or 2

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SCAN:DETEctor<x>:LIMit <arg>
```

```
SENSe:EMC:MEASurement:SCAN:DETEctor<x>:LIMit?
```

## Arguments

<arg> = Either ONE , TWO or THREE

ONE = Sets Limit to Limit 1

TWO = Sets Limit to Limit 2

THREE = Sets Limit to Limit 3

## Returns

<String>See Argument

## Examples

`SENSe:EMC:MEASurement:SCAN:DETEctor1:LIMit?` might return `TWO`, indicating that the limit selected for scan detector 1 is Limit 2.

## SENSe:EMC:MEASurement:SCAN:DWELI:TIME

Sets or queries the dwell time of scan detection in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:SCAN:DWELI:TIME <value>`

`SENSe:EMC:MEASurement:SCAN:DWELI:TIME?`

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

`SENSe:EMC:MEASUREMENT:SCAN:DWELL:TIME` might return `640.315E-3`, indicating the dwell time of scan detection is 640.3 ms.

## SENSe:EMC:MEASurement:SCAN:DWELI:TIME:AUTO

Sets or queries to automatically set the dwell time of scan detection in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:SCAN:DWELI:TIME:AUTO { 0 | OFF | 1 | ON }`

```
SENSe:EMC:MEASurement:SCAN:DWELL:TIME:AUTO?
```

### Arguments

0 or OFF disables the Auto check box of dwell time

1 or ON enables the Auto check box of dwell time

### Returns

<Boolean> 0,1

### Examples

SENSe:EMC:MEASUREMENT:SCAN:DWELL:TIME:AUTO might return 1, indicating that the dwell time of scan detection is set automatically.

## SENSe:EMC:MEASurement:SCAN:START (No Query Form)

Starts the scan on the signal with respect to selected scan detector and displays the measurement values.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:MEASurement:SCAN:START {SINGle | CONTInuous | STOP}
```

### Arguments

SINGle = Performs a Single scan

CONTInuous = Performs a Continuous scan

STOP = Stops the acquisition

### Returns

None

### Examples

SENSe:EMC:MEASUREMENT:SCAN:START SINGle performs a Single scan with respect to the selected scan detector.

## SENSe:EMC:MEASurement:SPOT:ALL

Sets or queries to enable or disable All check box to include all spots for spot analysis.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SPOT:ALL { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:SPOT:ALL?
```

## Arguments

0 or OFF disables the All check box

1 or ON enables the All check box

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASUREMENT:SPOT:ALL might return 1, indicating that all spots are included for spot analysis.

## SENSe:EMC:MEASurement:SPOT:COUNT

Sets or queries the Maximum number of spots required for spot Analysis in EMC View. Applies only to Pre-scan + Auto spot measurement type.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SPOT:COUNT <value>
```

```
SENSe:EMC:MEASurement:SPOT:COUNT?
```

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

`SENSE:EMC:MEASUREMENT:SPOT:COUNT` might return 20, indicating 20 spots to be selected for spot analysis.

## **SENSe:EMC:MEASurement:SPOT:DETEctor<x>**

Sets or queries the detector type of the specified spot detector in EMC View.

<x> = 1 or 2 or 3

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:SPOT:DETEctor<x> <arg>`

`SENSe:EMC:MEASurement:SPOT:DETEctor<x>?`

## Arguments

<arg> =

AVER = Sets Avg (VRMS) as the detector type

PEAK = Sets +Peak as the detector type

MILSTD = Sets Mil +Peak as the detector type

QUAS = Sets CISPR QPk as the detector type

CPE = Sets CISPR +Pk as the detector type

CAV = Sets CISPR Avg as the detector type

AVGL = Sets Avg (of logs) as the detector type

CAVL = Sets CISPR Avg (of logs) as the detector type

DSA = Sets DEF STAN Avg as the detector type

DSP = Sets DEF STAN Peak as the detector type

## Returns

<String> See Argument

## Examples

`SENSe:EMC:MEASurement:SPOT:DETEctor1?` might return QUAS, indicating that the spot detector type for detector 1 is CISPR QPk.

## **SENSe:EMC:MEASurement:SPOT:DETEctor<x>:ENABle**

Sets or queries the enabling and disabling of specified spot detector in EMC View.

<x> = 1 or 2 or 3

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:MEASurement:SPOT:DETEctor<x>:ENABle { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:SPOT:DETEctor<x>:ENABle?
```

### **Arguments**

0 or OFF disables the spot detector

1 or ON enables the spot detector

### **Returns**

<Boolean> 0,1

### **Examples**

SENSe:EMC:MEASurement:SPOT:DETEctor1:ENABle? might return 1, indicating that spot detector 1 is enabled.

## **SENSe:EMC:MEASurement:SPOT:DETEctor<x>:LIMit**

Sets or queries the limit of specified spot detector in EMC View.

<x> = 1 or 2 or 3

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:MEASurement:SPOT:DETEctor<x>:LIMit <arg>
```

```
SENSe:EMC:MEASurement:SPOT:DETEctor<x>:LIMit?
```

### **Arguments**

<arg> = Either ONE , TWO or THREE

ONE = Sets Limit to Limit 1  
TWO = Sets Limit to Limit 2  
THREE = Sets Limit to Limit 3

## Returns

<String> See Argument

## Examples

`SENSe:EMC:MEASurement:SPOT:DETEctor2:LIMit?` might return ONE, indicating that the limit selected for spot detector 2 is Limit 1.

## SENSe:EMC:MEASurement:SPOT:DWELI:TIME

Sets or queries the dwell time of spot detection in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:MEASurement:SPOT:DWELI:TIME <value>`

`SENSe:EMC:MEASurement:SPOT:DWELI:TIME?`

## Arguments

<value> = <NRf>

## Returns

<NRf>

## Examples

`SENSe:EMC:MEASUREMENT:SPOT:DWELL:TIME` might return 1.000000, indicating that the dwell time of spot detection is 1 s.

## SENSe:EMC:MEASurement:SPOT:DWELI:TIME:AUTO

Sets or queries to automatically set the dwell time of spot detection in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SPOT:DWELL:TIME:AUTO { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:MEASurement:SPOT:DWELL:TIME:AUTO?
```

## Arguments

0 or OFF disables the Auto check box of dwell time

1 or ON enables the Auto check box of dwell time

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:MEASUREMENT:SPOT:DWELL:TIME:AUTO might return 1, indicating that the dwell time of spot detection is set automatically.

## SENSe:EMC:MEASurement:SPOT:STARt (No Query Form)

Starts the spot re-measurement on the selected spots with respect to selected spot detector.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:MEASurement:SPOT:STARt
```

## Arguments

None

## Returns

None

## Examples

SENSe:EMC:MEASUREMENT:SPOT:STARt starts spot re-measurement on the selected spots.



## SENSe:EMC:MEASurement:TYPE

Sets or queries the measurement type in EMC View.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:MEASurement:TYPE {PSMS | PSAS | CS}
```

```
SENSe:EMC:MEASurement:TYPE?
```

### Arguments

PSMS = Sets Pre-scan+ Manual Spot as measurement type.

PSAS = Sets Pre-scan+ Auto Spot as measurement type.

CS = Sets Complete scan as measurement type.

### Returns

<String> PSMS or PSAS or CS

### Examples

SENSe:EMC:MEASUREMENT:TYPE might return CS, indicating that Complete scan is the measurement type selected.

## SENSe:EMC:MODE

Sets or queries the frequency ranges (mode) in EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:MODE {MULT | SING}
```

```
SENSe:EMC:MODE?
```

### Arguments

MULT = Sets multiple as frequency range

SING = Sets Single as frequency range

## Returns

<String> MULT or SING

## Examples

`SENSE:EMC:MODE` might return SING, indicating that the frequency range is Single.

## SENSe:EMC:OPTimization

Sets or queries the method of optimizing the gain and input bandwidth in the EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:OPTimization { AUTO | MINTime | MAXDynrange | MINNoise }
```

```
SENSe:EMC:OPTimization?
```

## Arguments

AUTO = optimizes automatically the gain and input bandwidth.

MINTime = optimizes the gain and input bandwidth to minimize sweep time.

MAXDynrange = optimizes the gain and input bandwidth to maximize the dynamic range.

MINNoise = optimizes the gain and input bandwidth to minimize noise.

## Returns

<String> See Argument

## Examples

`SENSe:EMC:OPTIMIZATION AUTO` optimizes automatically the gain and input bandwidth.

## SENSe:EMC:POINTs:COUNT

Sets or queries the trace points value in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:POINts:COUNt {P801 | P2401 | P4001 | P8001 | P10401 | P16001 | P32001 | P64001}
```

```
SENSe:EMC:POINts:COUNt?
```

## Arguments

P801 = Sets 801 as trace point value.

P2401 = Sets 2401 as trace point value.

P4001 = Sets 4001 as trace point value.

P8001 = Sets 8001 as trace point value.

P10401 = Sets 10401 as trace point value.

P16001 = Sets 16001 as trace point value.

P32001 = Sets 32001 as trace point value.

P64001 = Sets 64001 as trace point value.

## Returns

<String> See Arguments

## Examples

```
SENSe:EMC:POINTS:COUNT P2401 sets 2401 as trace point value in EMC view.
```

## SENSe:EMC:RANGe<x>:BANDwidth:VIDeo

Sets or queries the (Video Bandwidth) VBW value of selected range in EMC View.

<x> = <NR1> 1 to 20

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:RANGe<x>:BANDwidth:VIDeo <value>
```

```
SENSe:EMC:RANGe<x>:BANDwidth:VIDeo?
```

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

`SENSe:EMC:RANGe1:BANDwidth:VIDeo` might return `1.000000E+6`, indicating that the VBW value of range 1 is 1 MHz.

## **SENSe:EMC:RANGe<x>:BANDwidth:VIDeo:STATe**

Sets or queries the enabling or disabling the (Video state) VBW check box of specified range in EMC View.

<x> = <NR1> 1 to 20

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

`SENSe:EMC:RANGe<x>:BANDwidth:VIDeo:STATe { 0 | OFF | 1 | ON }`

`SENSe:EMC:RANGe<x>:BANDwidth:VIDeo:STATe?`

### Arguments

0 or OFF disables the VBW check box

1 or ON enables the VBW check box

### Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:RANGe2:BANDwidth:VIDeo:STATe` might return 0, indicating that VBW check box is disabled for range 2.

## **SENSe:EMC:RANGe<x>:EXCursion**

Sets or queries the excursion value of the specified range in EMC View.

<x> = <NR1> 1 to 20

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

`SENSe:EMC:RANGe<x>:EXCursion <value>`

SENSe:EMC:RANGe<x>:EXCURSION?

### Arguments

<value> = <NR3>

### Returns

<NR3>

### Examples

SENSe:EMC:RANGe1:EXCURSION might return 6.0000, indicating that the threshold value of range 1 is 6 dB.

## SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth

Sets or queries the (Resolution Bandwidth) RBW value of selected range in EMC View.

<x> = <NR1> 1 to 20

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth <value>

SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth?

### Arguments

<value> = <NR3>

### Returns

<NR3>

### Examples

SENSe:EMC:RANGe1:FREQuency:STOP might return 9.00, indicating that the RBW value of range 1 is 9 kHz.

## SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO

Sets or queries the Enabling or disabling of RBW Auto checkbox of selected range to set RBW value automatically in EMC View.

<x> = <NR1> 1 to 20

### Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO { 0 | OFF | 1 | ON}
```

```
SENSe:EMC:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO?
```

## Arguments

0 or OFF disables the Auto check box

1 or ON enables the Auto check box

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:RANGe0:FREQuency:STOP might return 0, indicating that Auto check box is disabled.

## SENSe:EMC:RANGe<x>:FREQuency:START

Sets or queries the Start frequency of the specified range in EMC View.

<x> = <NR1> 1 to 20

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:RANGe<x>:FREQuency:START <value>
```

```
SENSe:EMC:RANGe<x>:FREQuency:START?
```

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

SENSe:EMC:RANGe1:FREQuency:START might return 150000, indicating that the start frequency of range 1 is 150 kHz.

## SENSe:EMC:RANGe<x>:FREQuency:STOP

Sets or queries the stop frequency of selected range in EMC View.

<x> = <NR1> 1 to 20

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:RANGe<x>:FREQuency:STOP <value>
```

```
SENSe:EMC:RANGe<x>:FREQuency:STOP?
```

### Arguments

<value> = <NR3>

### Returns

<NR3>

### Examples

SENSe:EMC:RANGe1:FREQuency:STOP might return 500000, indicating that the stop frequency of range 1 is 500 kHz.

## SENSe:EMC:RANGe<x>:LIMit<y>:EQUAL

Sets or queries whether limit start and stop value of selected range and limit are the same in EMC View.

<x> = <NR1> 1 to 20

<y> = <NR1> Either Limit 1 or 2 or 3

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:RANGe<x>:LIMit<y>:EQUAL { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:RANGe<x>:LIMit<y>:EQUAL?
```

### Arguments

0 or OFF disables the Limit same check box

1 or ON enables the Limit same check box

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:RANGe1:LIMit1:EQUAL` might return 1, indicating the Limit 1 start and stop value for range 2 are the same.

## SENSe:EMC:RANGe<x>:LIMit<y>:START

Sets or queries the Limit start value of the specified range and specified limit in EMC View.

<x> = <NR1> 1 to 20

<y> = <NR1> Either Limit 1 or 2 or 3

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:RANGe<x>:LIMit<y>:START <value>`

`SENSe:EMC:RANGe<x>:LIMit<y>:START?`

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

`SENSe:EMC:RANGe1:LIMit1:START` might return 79.0000, indicating that the Limit 1 start value of range 1 is 79 dBuV.

## SENSe:EMC:RANGe<x>:LIMit<y>:STOP

Sets or queries the Limit stop value of the specified range and specified limit in EMC View.

<x> = <NR1> 1 to 20

<y> = <NR1> Either Limit 1 or 2 or 3

## Conditions

Measurement view: EMC - EMI Display



## Group

Sense commands

## Syntax

```
SENSe:EMC:RANGe<x>:LIMit<y>:STOP <value>
```

```
SENSe:EMC:RANGe<x>:LIMit<y>:STOP?
```

## Arguments

<value> = <NR3>

## Returns

<NR3>

## Examples

SENSe:EMC:RANGe1:LIMit1:STOP might return 79.0000, indicating that the Limit 1 stop value of range 1 is 79 dBuV.

## SENSe:EMC:RANGe<x>:STATe

Sets or queries Enabling and disabling the selected range in EMC View.

<x> = <NR1> 1 to 20

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:RANGe<x>:STATe { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:RANGe<x>:STATe?
```

## Arguments

0 or OFF disables the range

1 or ON enables the range

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:RANGe2:STATe might return 0, indicating that range 2 is disabled.

## **SENSe:EMC:RANGe<x>:THReshold**

Sets or queries the threshold value of the specified range in EMC View.

<x> = <NR1> 1 to 20

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:RANGe<x>:THReshold <value>
```

```
SENSe:EMC:RANGe<x>:THReshold?
```

### **Arguments**

<value> = <NR3>

### **Returns**

<NR3>

### **Examples**

SENSe:EMC:RANGe1:THReshold might return 46.0000, indicating that the threshold value of range 1 is 46 dBuV.

## **SENSe:EMC:REGion**

Sets or queries the region of standard in EMC view.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:REGion {EURO | US}
```

```
SENSe:EMC:REGion?
```

### **Arguments**

EURO = Sets Europe as the region

US = Sets US as the region

## Returns

<String> See Argument

## Examples

`SENSe:EMC:REGION` might return US, indicating that the region selected is US.

## SENSe:EMC:REPOrt:INCLude:CONTRol:SETTings

Sets or queries to include or exclude control settings in report by enabling or disabling include control settings check box in report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:REPOrt:INCLude:CONTRol:SETTings { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:REPOrt:INCLude:CONTRol:SETTings?
```

## Arguments

0 or OFF exclude the control settings in report

1 or ON include the control settings in report

## Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:REPORT:INCLUDE:CONTROL:SETTINGS` might return 1, indicating that control settings is included in report.

## SENSe:EMC:REPOrt:INCLude:EMISsion:RESUlts

Sets or queries to include or exclude emission results in report by enabling or disabling include emission results check box in report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:REPOrt:INCLude:EMISSion:RESUlts { 0 | OFF | 1 | ON}
```

```
SENSe:EMC:REPOrt:INCLude:EMISSion:RESUlts?
```

## Arguments

0 or OFF exclude emission results in report

1 or ON include emission results in report

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:REPORT:INCLUDE:EMISSION:RESULTS might return 1, indicating that emission results is included in report.

## SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation

Sets or queries to include or exclude environment information in report by enabling or disabling environment information check box in report setup of EMC display.

<x> = <NR1> 1 to 20

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation { 0 | OFF | 1 | ON}
```

```
SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation?
```

## Arguments

0 or OFF exclude environment information in report

1 or ON include environment information in report

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:REPORT:INCLUDE:ENVIRONMENT:INFORMATION might return 1, indicating that environment information is included in report.

---

## **SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation:ADD (No Query Form)**

Adds one Environment information content by adding one title and its corresponding value in the report setup of EMC display.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation:ADD "Title","Value"
```

### **Arguments**

Title = <String>

Value = <String>

### **Returns**

None

### **Examples**

`SENSe:EMC:REPORT:INCLUDE:ENVIRONMENT:INFORMATION:ADD "Temperature","20"` adds the environment information temperature and its value as 20 in the report.

## **SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation:DELeTe:ALL (No Query Form)**

Deletes all the environment information in the report setup of EMC display.

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Sense commands

### **Syntax**

```
SENSe:EMC:REPOrt:INCLude:ENVIronment:INFOrmation:DELeTe:ALL
```

### **Arguments**

None

### **Returns**

None

## Examples

`SENSe:EMC:REPORT:INCLUDE:ENVIRONMENT:INFORMATION:DELETE:ALL` deletes all the environment information in the report.

## SENSe:EMC:REPOrt:INCLude:EUT:INFOrmation

Sets or queries to include or exclude EUT (Equipment under test) information in report by enabling or disabling EUT information check box in report setup of EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:REPOrt:INCLude:EUT:INFOrmation { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:REPOrt:INCLude:EUT:INFOrmation?
```

### Arguments

0 or OFF exclude EUT information in report

1 or ON include EUT information in report

### Returns

<Boolean> 0,1

## Examples

`SENSe:EMC:REPORT:INCLUDE:EUT:INFORMATION` might return 0, indicating that EUT information is excluded in report.

## SENSe:EMC:REPOrt:INCLude:EUT:INFOrmation:ADD (No Query Form)

Adds one EUT information content by adding one title and its corresponding value in the report setup of EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:REPOrt:INCLude:EUT:INFOrmation:ADD "Title","Value"
```

## Arguments

Title = <String>

Value = <String>

## Returns

None

## Examples

`SENSe:EMC:REPORT:INCLUDE:EUT:INFORMATION:ADD "Serial Number","ABC156"` adds the EUT information Serial Number and its value as ABC156 in the report.

## **SENSe:EMC:REPOrt:INCLude:EUT:INFORmation:DELeTe:ALL (No Query Form)**

Deletes all the EUT information in the report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:REPOrt:INCLude:EUT:INFORmation:DELeTe:ALL`

## Arguments

None

## Returns

None

## Examples

`SENSe:EMC:REPORT:INCLUDE:EUT:INFORMATION:DELETE:ALL` deletes all the EUT information in the report.

## **SENSe:EMC:REPOrt:INCLude:GRAPh**

Sets or queries to include or exclude graph of measurement results in report by enabling or disabling include graph check box in report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:REPOrt:INCLude:GRAPh { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:REPOrt:INCLude:GRAPh?
```

## Arguments

0 or OFF exclude the graph in report

1 or ON include the graph in report

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:REPORT:INCLUDE:GRAPH might return 0, indicating that graph is excluded in report.

## SENSe:EMC:REPOrt:INCLude:INSPect:FREQuency:RESUlts

Sets or queries to include or exclude inspect frequency results in report by enabling or disabling include inspect req results check box in report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

```
SENSe:EMC:REPOrt:INCLude:INSPect:FREQuency:RESUlts { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:REPOrt:INCLude:INSPect:FREQuency:RESUlts?
```

## Arguments

0 or OFF exclude inspect frequency results in report

1 or ON include inspect frequency results in report

## Returns

<Boolean> 0,1

## Examples

SENSe:EMC:REPORT:INCLUDE:INSPECT:FREQUENCY:RESULTS might return 0, indicating that inspect frequency results is excluded in report.



---

## SENSe:EMC:REPOrt:INCLude:USER:INFOrmation

Sets or queries to include or exclude user information in report by enabling or disabling user information check box in report setup of EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:REPOrt:INCLude:USER:INFOrmation { 0 | OFF | 1 | ON }
```

```
SENSe:EMC:REPOrt:INCLude:USER:INFOrmation?
```

### Arguments

0 or OFF exclude user information in report

1 or ON include user information in report

### Returns

<Boolean> 0,1

### Examples

SENSe:EMC:REPORT:INCLUDE:USER:INFORMATION might return 1, indicating that user information is included in report.

## SENSe:EMC:REPOrt:INCLude:USER:INFOrmation:ADD (No Query Form)

Adds one user information content by adding one title and its corresponding value in the report setup of EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Sense commands

### Syntax

```
SENSe:EMC:REPOrt:INCLude:USER:INFOrmation:ADD "Title","Value"
```

### Arguments

Title = <String>

Value = <String>

## Returns

None

## Examples

`SENSe:EMC:REPORT:INCLUDE:USER:INFORMATION:ADD "Engineer","xyz"` adds the user information engineer and its value as xyz in the report.

## **SENSe:EMC:REPOrt:INCLude:USER:INFOrmation:DELeTe:ALL (No Query Form)**

Deletes all the user information in the report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:REPOrt:INCLude:USER:INFOrmation:DELeTe:ALL`

## Arguments

None

## Returns

None

## Examples

`SENSe:EMC:REPORT:INCLUDE:USER:INFORMATION:DELETE:ALL` deletes all the user information in the report.

## **SENSe:EMC:REPOrt:MEASurement:HEADing**

Sets or queries the measurement heading in the report setup of EMC display.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:REPOrt:MEASurement:HEADing <value>`

`SENSe:EMC:REPOrt:MEASurement:HEADing?`

## Arguments

<value> = <String>

## Returns

<String>

## Examples

`SENSe:EMC:REPORT:MEASUREMENT:HEADING` might return "spot", indicating that the measurement heading of the report is spot.

## SENSe:EMC:STANdard

Sets or queries the standard in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Sense commands

## Syntax

`SENSe:EMC:STANdard <arg>`

`SENSe:EMC:STANdard?`

## Arguments

When the region is US

<arg> = {FCC Part 15 | FCC Part 18 | MIL-STD 461G}

When the region is Europe

<arg> = {EN55011 | EN55013| EN55014 | EN55015 | EN55025 | EN55032 |EN60601 | DEF STAN}

## Returns

<String> See Argument

## Examples

`SENSe:EMC:STANDARD FCC Part 15` sets FCC Part 15 as the standard.

## [SENSe]:FDVTime[:MEASview<y>]:SELEcted:SOURce

Sets or queries the source of Frequency deviation versus Time for a selected measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

The source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:FDVTime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"  
| "Source 4"}
```

```
[SENSe]:FDVTime[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents the measurement index based on the number of sources of connected instrument, where the MEASview keyword is optional by default and y is 1.

## Examples

`SENSe:FDVTime:MEASview<y>:SELECTED:SOURce?` queries the Source for Frequency deviation versus Time measurement view 1.

## [SENSe]:{FM|PM}:BURSt:THReshold

Sets or queries the threshold level above which the input signal is determined to be a burst in the FM/PM measurement. The burst detected first is used for the measurement.

## Conditions

Measurement views: FM, PM

## Group

Sense commands

## Syntax

```
[SENSe]:{FM|PM}:BURSt:THReshold <value>
```

```
[SENSe]:{FM|PM}:BURSt:THReshold?
```

## Arguments

<value>: : <NRf> specifies the threshold level. Range: -100.0 to -10.0 dB.

## Returns

<Boolean> 0,1

## Examples

`SENSE:FM:BURST:THRESHOLD -10` sets the threshold level to -10 dB in the FM measurement.

## [SENSe]:{FM|PM}:FREQuency:OFFSet

Sets or queries the carrier frequency offset in the FM/PM measurement.

### Conditions

Measurement views: FM, PM

### Group

Sense commands

### Syntax

`[SENSe]:{FM|PM}:FREQuency:OFFSet <value>`

`[SENSe]:{FM|PM}:FREQuency:OFFSet?`

### Arguments

`<value>`: `<Nrf>` specifies the frequency offset from the center frequency. Range:  $-(\text{MeasBW} * 1.1) / 2$  to  $+(\text{MeasBW} * 1.1) / 2$  where MeasBW is set by the command `[SENSe]:{AM|FM|PM}:{BANDwidth|BWIDth}:MEASurement`.

## Examples

`SENSE:FM:FREQUENCY:OFFSET 10MHZ` sets the carrier frequency offset to 10 MHz.

## [SENSe]:{FM|PM}:FREQuency:OFFSet:MARKer (No Query Form)

Sets the frequency offset from the selected marker location in the FM measurement.

Sets the frequency offset from the selected delta marker location in the PM measurement.

### Conditions

Measurement views: FM, PM

### Group

Sense commands

### Syntax

`[SENSe]:{FM|PM}:FREQuency:OFFSet:MARKer`

### Arguments

None

## Examples

`SENSE:FM:FREQUENCY:OFFSET:MARKER` sets frequency offset from the selected Marker position.

## **[SENSe]:{FM|PM}:FREQuency:SEARch:AUTO**

Determines whether to detect the carrier frequency automatically or manually in the FM/PM measurements.

### Conditions

Measurement views: FM, PM

### Group

Sense commands

### Syntax

```
[SENSe]:{FM|PM}:FREQuency:SEARch:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:{FM|PM}:FREQuency:SEARch:AUTO?
```

### Arguments

`OFF` or `0` specifies that the carrier frequency is set manually. Use the command `[SENSe]:{FM|PM}:FREQuency:OFFSet` to set the carrier offset frequency.

`ON` or `1` specifies that the carrier frequency is detected automatically.

## Examples

`SENSE:FM:FREQUENCY:SEARCH:AUTO ON` specifies that the carrier frequency is detected automatically.

## **[SENSe]:{FSETtling|PSETtling}:{BANDwidth|BWIDTh}**

Sets or queries the measurement bandwidth for the Settling Time measurements.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Sense commands

### Syntax

```
[SENSe]:{FSETtling|PSETtling}:{BANDwidth|BWIDTh} <value>
```

```
[SENSe]:{FSETtling|PSETtling}:{BANDwidth|BWIDTh}?
```

### Related Commands

`FETCh:{AM|FM|PM}?`

`READ:AM:RESult?`

## Arguments

<value>::<NRf> specifies the bandwidth for the specified settling time measurement.

## Examples

`SENSE:FSETTLING:BANDwidth 1e6` sets the Frequency Settling time measurement bandwidth to 1 MHz.

## [SENSe]:{FSETtling|PSETtling}:{BANDwidth|BWIDTh}:ACTual? (Query Only)

Queries the actual measurement bandwidth in the Settling Time measurements.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

`[SENSe]:{FSETtling|PSETtling}:{BANDwidth|BWIDTh}:ACTual?`

## Arguments

None

## Returns

<NRf> The actual measurement bandwidth in Hz.

## Examples

`SENSE:FSETTLING:BANDWIDTH:ACTUAL?` might return `40.000E+6`, indicating that the actual RBW is 40.00 MHz.

## [SENSe]:{FSETtling|PSETtling}:FREQuency:CENTer

Sets or queries the measurement frequency.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

`[SENSe]:{FSETtling|PSETtling}:FREQuency:CENTer <value>`

`[SENSe]:{FSETtling|PSETtling}:FREQuency:CENTer?`

## Arguments

<value>::<NRf> specifies the measurement frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:PSETTLING:FREQUENCY:CENTER 7.5GHz` sets the Phase Settling Time measurement frequency to 7.5 GHz.

## [SENSe]:{FSETtling|PSETtling}:FREQuency:OFFSet

Sets or queries the offset frequency.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

`[SENSe]:{FSETtling|PSETtling}:FREQuency:OFFSet <value>`

`[SENSe]:{FSETtling|PSETtling}:FREQuency:OFFSet?`

## Arguments

<value>::<NRf> specifies the offset frequency.

Range:  $\pm (0.5 \times \text{Measurement bandwidth})$

Default: 0 Hz.

## Examples

`SENSe:PSETTLING:FREQUENCY:OFFSET 20E6` sets the Phase Settling Time offset frequency to 20 MHz.

## [SENSe]:{FSETtling|PSETtling}:LENGth

Sets or queries the measurement length.

## Conditions

Measurement views: Frequency and Phase Settling Time



## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:LENGth <value>
```

```
[SENSe]:{FSETtling|PSETtling}:LENGth?
```

## Arguments

<value> : <NRf> specifies the measurement length in seconds.

Range: 10 ns to [(acquisition length) - 400 ns].

If [(analysis start) + (analysis length)] > [(acquisition length) - 400 ns], the actual measurement length is reduced to [(acquisition length) - 200 ns].

Default: 1 ms.

## Examples

`SENSe:PSETTLING:LENGTH 25.625us` sets the measurement length to 25.625  $\mu$ s.

## [SENSe]:{FSETtling|PSETtling}:LENGth:ACTual? (Query Only)

Queries the actual measurement length for the Settling Time measurements.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:LENGth:ACTual?
```

## Arguments

None

## Returns

<NRf> The actual measurement length in Hz.

## Examples

`SENSe:FSETTLING:LENGTH:ACTUAL?` might return 25.625E-6, indicating that the actual measurement length is 25.625  $\mu$ s.

## [SENSe]:{FSETtling|PSETtling}:MASK:LIMit

Sets or queries the frequency or phase limit for the specified region.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:MASK:LIMit <x> <value>
```

```
[SENSe]:{FSETtling|PSETtling}:MASK:LIMit <x>?
```

## Arguments

<x> ::= { 1 | 2 | 3 } specifies the mask region.

<value> ::= <Nrf> specifies the frequency or phase limit for the region.

For Frequency Settling Time:

- Range: 1 Hz to 1 GHz
- Default: 500 kHz

For Phase Settling Time:

- Range: 0.1 degree to 180 degrees
- Default: 1 degree

## Examples

`SENSe:FSETTLING:MASK:LIMIT2 7.5E5` sets the limit for region 2 to 750 kHz.

## [SENSe]:{FSETtling|PSETtling}:MASK:STATe

Sets or queries mask test state.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:MASK:STATe { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 enables Mask test.

ON or 1 disables Mask test.

## Examples

`SENSe:FSETTLING:MASK:STATE ON` enables mask test.

## [SENSe]:{FSETtling|PSETtling}:MASK:TIME:REFeRence

Sets or queries the mask time reference.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Sense commands

### Syntax

```
[SENSe]:{FSETtling|PSETtling}:MASK:TIME:REFeRence { TRIGger | TRANsition }
```

### Arguments

`TRIGger` specifies that the trigger point serve as the mask time reference.

`TRANsition` specifies that the point where the signal transitions out of the previous tolerance range serves as the mask time reference.

Default: Trigger

## Examples

`SENSe:FSETTLING:MASK:TIME:REFeRence TRIGGER` specifies that the trigger point serve as the mask time reference.

## [SENSe]:{FSETtling|PSETtling}:MASK:TIME:STARt

Sets or queries the mask region start time in seconds.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Sense commands

### Syntax

```
[SENSe]:{FSETtling|PSETtling}:MASK:TIME:STARt <x> <value>
```

```
[SENSe]:{FSETtling|PSETtling}:MASK:TIME:STARt <x>?
```

### Arguments

`<x> ::= { 1 | 2 | 3 }` specifies the mask region.

`<value> ::= <NRf>` specifies the start time of the mask region in seconds.

## Examples

`SENSe:FSETTLING:MASK:TIME:START3?` might return 1.0000E-3 as the start time for mask region 3.

## **[SENSe]:{FSETtling|PSETtling}:MASK:TIME:STOP**

Sets or queries the mask region stop time in seconds.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Sense commands

### Syntax

```
[SENSe]:{FSETtling|PSETtling}:MASK:TIME:STOP <value>
```

```
[SENSe]:{FSETtling|PSETtling}:MASK:TIME:STOP <x>?
```

### Arguments

`<value>::<NRf>` specifies the stop time of the mask in seconds.

## Examples

`SENSe:FSETTLING:MASK:TIME:STOP?` might return 6.244120000E-3 as the mask stop time.

## **[SENSe]:{FSETtling|PSETtling}:MAXTracepoints**

Sets or queries the maximum trace points in the Settling Time measurements.

### Conditions

Measurement views: Frequency and Phase Settling

### Group

Sense commands

### Syntax

```
[SENSe]:{FSETtling|PSETtling}:MAXTracepoints { ONEK | TENK | HUNDredk |  
NEVerdecimate }
```

```
[SENSe]:{FSETtling|PSETtling}:MAXTracepoints?
```

### Arguments

`ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

## Examples

SENSe:PSETTLING:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k.

## [SENSe]:{FSETtling|PSETtling}:SDURation:MINimum

Sets or queries the minimum settled duration in seconds. This duration time determines the result of :FETCh: {FSETtling | PSETtling}:SLMSd[:PASS]?

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:SDURation:MINimum <value>
```

```
[SENSe]:{FSETtling|PSETtling}:SDURation:MINimum?
```

## Arguments

<value>::<Nrf> specifies the minimum settled duration in seconds.

## Examples

SENSe:FSETTLING:SDURATION:MINIMUM 17.73E-3 sets the minimum settled duration to 17.73 ms.

## [SENSe]:{FSETtling|PSETtling}:TARGet:REFerence

Sets or queries the target reference frequency.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Sense commands

## Syntax

```
[SENSe]:{FSETtling|PSETtling}:TARGet:REFerence { AUTO | MFReq }
```

```
[SENSe]:{FSETtling|PSETtling}:TARGet:REFerence?
```

## Arguments

AUTO causes the instrument to determine the target reference value by averaging at least the last 256 samples at the end of the measurement period.

MFREQ specifies that the Measurement Frequency value is used as the target reference frequency.

Default: AUTO

## Examples

SENSe:FSETTLING:TARGET:REFERENCE MFREQ specifies that the target reference frequency is the measurement frequency.

## [SENSe]:{FSETtling|PSETtling}:TOLerance

Sets or queries the frequency or phase range into which the signal must settle.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Sense commands

### Syntax

[SENSe]:{FSETtling|PSETtling}:TOLerance <value>

[SENSe]:{FSETtling|PSETtling}:TOLerance?

### Arguments

<value>::<Nrf> specifies the frequency or phase range.

For Frequency Settling Time:

- Range: 1 Hz to 1 GHz
- Default: 500 kHz

For Phase Settling Time:

- Range: 0.1 degree to 180 degrees
- Default: 1 degree

### Examples

SENSe:FSETTLING:TOLERANCE 2.5E5 sets the tolerance to 250 kHz.

## [SENSe]:FVTime[:MEASview<y>]:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

### Conditions

Measurement views: Frequency versus Time

### Group

Sense commands

## Syntax

```
[SENSe]:FVTime[:MEASview<y>]:CLEAr:RESults
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

SENSe:FVTIME:MEASview1:CLEAR:RESULTS restarts multi-trace functions.

## [SENSe]:FVTime[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the Frequency versus Time measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:CENTer <value>
```

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:CENTer?
```

## Related Commands

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:START,
```

```
[SENSe]:FVTime:MEASview<y>:FREQuency:STOP
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

SENSe:FVTIME:MEASview1:FREQUENCY:CENTER 7.5GHz sets the center frequency to 7.5 GHz.

## **[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN**

Sets or queries the frequency span in the Frequency versus Time measurement.

### **Conditions**

Measurement views: Frequency versus Time

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN <value>
```

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> is the frequency span.

Range: 10 Hz to 40 MHz (Standard) / 110MHz (Option 110)

### **Examples**

`SENSe:FVTIME:MEASview1:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

## **[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN:LINK**

Sets or queries if frequency span is linked or not in the Frequency versus Time measurement.

### **Conditions**

Measurement views: Frequency versus Time

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN:LINK <value>
```

```
[SENSe]:FVTime[:MEASview<y>]:FREQuency:SPAN:LINK?
```

### **Arguments**

OFF or 0 specifies that Link to Span is not checked.

ON or 1 specifies that Link to Span is checked .



## Examples

`SENSe:FVTime:MEASview1:FREQUENCY:SPAN:LINK?` Might return 0 or 1 based on condition of Link to Span checkbox.

## [SENSe]:FVTime[:MEASview<y>]:FREQUENCY:START

Sets or queries the measurement start frequency (left edge on the graph) in the Frequency versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:CENTer` command.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

`[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:START <value>`

`[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:START?`

## Related Commands

`[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STOP`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value 1.

`<value>` : : `<Nrf>` is the measurement start frequency.

Range: (center frequency)  $\pm$  (span)/2.

## Examples

`SENSe:FVTIME:MEASview1:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

## [SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets `[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO OFF`.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP <value>
[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP?
```

## Related Commands

```
[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

SENSe:FVTIME:MEASview1:FREQUENCY:STEP 1.5kHz sets the step size to 1.5 kHz.

## [SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO { OFF | ON | 0 | 1 }
[SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the frequency step size is set manually using the [SENSe]:FVTime[:MEASview<y>]:FREQUENCY:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

## Examples

`SENSe:FVTIME:MEASview1:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

## [SENSe]:FVTime[:MEASview<y>]:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the Frequency versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:FVTime[:MEASview<y>]:FREQuency:CENTer` command.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

`[SENSe]:FVTime[:MEASview<y>]:FREQuency:STOP <value>`

`[SENSe]:FVTime[:MEASview<y>]:FREQuency:STOP?`

## Related Commands

`[SENSe]:FVTime[:MEASview<y>]:FREQuency:START`

## Arguments

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value 1.

`<value>` : : `<NRf>` is the measurement stop frequency.

Range: (center frequency)  $\pm$  (span)/2.

## Examples

`SENSe:FVTIME:MEASview1:FREQUENCY:STOP 7.05GHz` sets the stop frequency to 7.05 GHz.

## [SENSe]:FVTime[:MEASview<y>]:MAXTracepoints

Sets or queries the maximum trace points in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:FVTime[:MEASview<y>]:MAXTracepoints { ONEK | TENK | HUNDredk |  
NEVerdecimate }
```

```
[SENSe]:FVTime[:MEASview<y>]:MAXTracepoints?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

ONEK sets the maximum trace points to 1 k.

TENK sets the maximum trace points to 10 k.

HUNDredk sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

## Examples

```
SENSe:FVTIME:MEASview1:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k.
```

## [SENSe]:FVTime[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source with respect to the Frequency versus Time Measurement View number

## Conditions

Measurement views: Frequency versus Time with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
SENSe:FVTime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"  
| "Source 4"}
```

```
SENSe:FVTime[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

```
SENSe:FVTime:MEASview1:SELEcted:SOURce? queries or sets the Source for Frequency versus Time Measurement View 1.
```

## [SENSe]:IQVTime[:MEASview<y>]:CLEAr:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

### Conditions

Measurement views: RF I&Q versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:CLEAr:RESults
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Examples

SENSe:IQVTIME:MEASview1:CLEAR:RESULTS restarts multi-trace functions.

## [SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the RF I&Q versus Time measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer <value>
```

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer?
```

### Related Commands

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:START,  
[SENSe]:IQVTime:MEASview<y>:FREQuency:STOP
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:IQVTIME:MEASview1:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

## [SENSe]:IQVTime[:MEASview<y>]:FREQUENCY:SPAN

Sets or queries the frequency span in the RF I&Q versus Time measurement.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Sense commands

### Syntax

`[SENSe]:IQVTime[:MEASview<y>]:FREQUENCY:SPAN <value>`

`[SENSe]:IQVTime[:MEASview<y>]:FREQUENCY:SPAN?`

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional.  
Default value 1.

<value>::<NRf> is the frequency span.

Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

## Examples

`SENSe:IQVTIME:MEASview1:FREQUENCY:SPAN 20MHz` sets the span to 20 MHz.

## [SENSe]:IQVTime[:MEASview<y>]:FREQUENCY:SPAN:LINK

Sets or queries if frequency span is linked or not in the RF I & Q versus Time measurement.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Sense commands

## Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:SPAN:LINK <value>
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:SPAN:LINK?
```

## Arguments

OFF or 0 specifies that Link to Span is not checked.

ON or 1 specifies that Link to Span is checked .

## Examples

SENSe:IQVTime:MEASview1:FREQuency:SPAN:LINK? Might return 0 or 1 based on condition of Link to Span checkbox.

## [SENSe]:IQVTime[:MEASview<y>]:FREQuency:START

Sets or queries the measurement start frequency (left edge on the graph) in the RF I&Q versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer command.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:START <value>
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:START?
```

## Related Commands

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STOP
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> is the measurement start frequency.

Range: (center frequency) ± (span)/2.

## Examples

SENSe:IQVTIME:MEASview1:FREQUENCY:START 6.95GHz sets the start frequency to 6.95 GHz.

## [SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO OFF.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP <value>
```

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP?
```

### Related Commands

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<Nrf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

### Examples

```
SENSe:IQVTIME:MEASview1:FREQUENCY:STEP 1.5kHz sets the step size to 1.5 kHz.
```

## [SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

### Conditions

Measurement views: RF I&Q versus Time

### Group

Sense commands

### Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
```



```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP:AUTO?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the frequency step size is set manually using the  
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

## Examples

SENSe:IQVTIME:MEASview1:FREQUENCY:STEP:AUTO ON sets the frequency step size automatically.

## [SENSe]:IQVTime[:MEASview<y>]:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the RF I&Q versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the  
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:CENTer command.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STOP <value>
```

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:STOP?
```

## Related Commands

```
[SENSe]:IQVTime[:MEASview<y>]:FREQuency:START
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<Nrf> is the measurement stop frequency.

Range: (center frequency) ± (span)/2.

## Examples

SENSe:IQVTIME:MEASview1:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.

## **[SENSe]:IQVTime[:MEASview<y>]:MAXTracepoints**

Sets or queries the maximum trace points in the RF I&Q versus Time measurement.

### **Conditions**

Measurement views: RF I&Q versus Time

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:IQVTime[:MEASview<y>]:MAXTracepoints { ONEK | TENK | HUNDredk |  
NEVerdecimate }
```

```
[SENSe]:IQVTime[:MEASview<y>]:MAXTracepoints?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

ONEK sets the maximum trace points to 1 k.

TENK sets the maximum trace points to 10 k.

HUNDredk sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

### **Examples**

```
SENSe:IQVTIME:MEASview1:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k.
```

## **[SENSe]:IQVTime[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source with respect to the RF I&Q versus Time Measurement View number

### **Conditions**

Measurement views: RF I&Q versus Time

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### **Group**

Sense commands

### **Syntax**

```
SENSe:IQVTime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"  
| "Source 4"}
```

```
SENSe:IQVTime[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

```
SENSe:IQVTime:MEASview1:SELEcted:SOURce? queries or sets the Source for RF I&Q versus Time.
```

## SENSe:LTE:ACLR:BANDwidth:RESolution

Sets or queries the resolution bandwidth (RBW) in the LTE ACLR display.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:BANDwidth:RESolution
```

## Arguments

<value>::<NRf> specifies the resolution bandwidth

## Returns

<NR3>

## Examples

```
SENSe:LTE:ACLR:BANDwidth:RESOLUTION? might return 100.0000000E+3, indicating the RBW is 100 kHz.
```

## SENSe:LTE:ACLR:BANDwidth:VIDeo

Sets or queries the video bandwidth (VBW) in the LTE ACLR display.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:BANDwidth:VIDeo <value>
```

## Arguments

<value>::<NRf> specifies the video bandwidth

## Returns

<NR3>

## Examples

`SENSe:LTE:ACLR:BANDWIDTH:VIDEO 2.000000E+3` sets the VBW to 2 kHz.

## SENSe:LTE:ACLR:BANDwidth:VIDeo:STATe

Sets or queries enabling or disabling the video bandwidth (VBW) in the LTE ACLR display.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

`SENSe:LTE:ACLR:BANDwidth:VIDeo:STATe { 0 | 1 }`

`SENSe:LTE:ACLR:BANDwidth:VIDeo:STATe?`

## Arguments

0 disables the VBW.

1 enables the VBW.

## Returns

0 means that the VBW is disabled.

1 means that the VBW is enabled.

## Examples

`SENSe:LTE:ACLR:BANDWIDTH:VIDEO:STATE 1` sets the VBW to automatic (enabled) and is decided by the analysis module.

## SENSe:LTE:ACLR:CHANnel:IBANdwidth

Sets or queries the reference channel integration bandwidth for the LTE ACLR display.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:CHANnel:IBANdwidth <value>
```

```
SENSe:LTE:ACLR:CHANnel:IBANdwidth?
```

## Arguments

<value>::<NRf> specifies the channel integration bandwidth.

## Returns

<NR3>

## Examples

SENSe:LTE:ACLR:CHANNEL:IBANDWIDTH? might return 18.000000E+6 indicating that the integration bandwidth is 18 MHz.

## SENSe:LTE:ACLR:CHANnel:WIDTh

Sets or queries the reference channel width for the LTE ACLR display.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:CHANnel:WIDTh <value>
```

```
SENSe:LTE:ACLR:CHANnel:WIDTh?
```

## Arguments

<value>::<NRf> specifies the reference channel width.

## Returns

<NR3>

## Examples

SENSe:LTE:ACLR:CHANNEL:WIDTH 5.00000000E+6 sets the reference channel width to 5 MHz.

## SENSe:LTE:ACLR:CLEar:RESults (No Query Form)

Clears the results in the LTE ACLR measurement.

## Conditions

Measurement views: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:CLEAr:RESults
```

## Arguments

None

## Examples

`SENSe:LTE:ACLR:CLEAR:RESULTS` clears the results of the LTE ACLR measurement.

## SENSe:LTE:ACLR:FREQuency

Sets or queries the measurement center frequency in the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:FREQuency <value>
```

```
SENSe:LTE:ACLR:FREQuency?
```

## Arguments

`<value>::<NRf>` specifies the center frequency

## Returns

`<NR3>`

## Examples

`SENSe:LTE:ACLR:FREQUENCY 1.000000000E+9` sets the center frequency to 1.0 GHz.

## SENSe:LTE:ACLR:MFILter

Sets or queries the shape of the filter for the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:MFILter { GAUSSian | RECTangular }
SENSe:LTE:ACLR:MFILter?
```

## Arguments

GAUSSian: sets the filter shape to Gaussian.

RECTangular: sets the filter shape to rectangular.

## Returns

GAUSSIAN: means the filter shape is Gaussian.

RECTangular: means the filter shape is rectangular.

## Examples

```
SENSe:LTE:ACLR:MFILTER RECT sets the filter shape to rectangular.
```

## SENSe:LTE:ACLR:MODE:REALtime

Sets or queries the real-time LTE ACLR measurement mode. When set to real-time, the reference channel and all requested adjacent channel regions are acquired together. When real-time if off, the measurement is done in Swept mode (an acquisition for each adjacent band).

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:MODE:REALtime { OFF | ON | 0 | 1 }
SENSe:LTE:ACLR:MODE:REALtime?
```

## Arguments

OFF or 0 turns off the automatic update.

ON or 1 turns on automatic updates center frequency.

## Returns

OFF or 0 means automatic update is off.

ON or 1 means automatic updates are off.

## Examples

`SENSe:LTE:ACLR:MODE:REALTIME?` might return 1 to indicate the real-time checkbox in the LTE ACLR control panel is enabled.

## **SENSe:LTE:ACLR:NFLoor:ACquire (No Query Form)**

Enables measurement of the noise floor for the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

`SENSe:LTE:ACLR:NFLoor:ACquire`

## Arguments

None

## Examples

`SENSe:LTE:ACLR:NFLoor:ACQUIRE` will enable measurement of the noise floor in the LTE ACLR measurement.

## **SENSe:LTE:ACLR:NFLoor:CORRected? (Query Only)**

Returns whether or not noise correction was applied on the last acquisition in the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

`SENSe:LTE:ACLR:NFLoor:CORRected?`

## Returns

1 means that noise correction was applied on the last acquisition.



0 means that noise correction was not applied on the last acquisition.

## Examples

`SENSe:LTE:ACLR:NFLoor:CORRECTED?` might return 1, indicating that noise correction was applied on the last acquisition.

## SENSe:LTE:ACLR:NFLoor:STATe

Determines whether to enable or disable the correction for noise floor for the LTE ACLR measurement.

### Conditions

Measurement view: LTE ACLR

### Group

Sense commands

### Syntax

```
SENSe:LTE:ACLR:NFLoor:STATe { OFF | ON | 0 | 1 }
```

```
SENSe:LTE:ACLR:NFLoor:STATe?
```

### Arguments

OFF or 0 disables noise correction in the measurement view.

ON or 1 enables noise correction in the measurement view.

### Returns

OFF or 0 means that noise correction is disabled in the measurement view.

ON or 1 means that noise correction is enabled in the measurement view.

## Examples

`SENSe:LTE:ACLR:NFLoor:STATe OFF` disables noise correction in the measurement view.

## SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution

Sets or queries the resolution bandwidth (RBW) value in the specified row in the Offsets & Limits table for the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

### Group

Sense commands

## Syntax

SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution <value>

SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution?

## Arguments

<value>::<NRf> specifies the RBW in the specified row.

## Returns

<NR3>

## Examples

SENSe:LTE:ACLR:OFFSET4:BANDWIDTH:RESOLUTION 30.00000000E+3 sets the RBW to 30 kHz for row 4.

## SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:COUNT

Sets or queries the count for the resolution bandwidths in the specified row in the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

## Group

Sense commands

## Syntax

SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:COUNT <value>

SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:COUNT?

## Arguments

<value>::<NR1> specifies a count integer

## Returns

<NR1>

## Examples

SENSe:LTE:ACLR:OFFSET1:BANDWIDTH:RESOLUTION:COUNT 10 sets the count for row 1 to 10.

## SENSe:LTE:ACLR:OFFSet<X>:BANDwidth:RESolution:FILTer

Sets or queries the filter bandwidth resolution under Offset & Limit table tab for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

Parameter <x> represents the row in the Offsets & Limits table

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<X>:BANDwidth:RESolution:FILTer <value>
```

```
SENSe:LTE:ACLR:OFFSet<X>:BANDwidth:RESolution:FILTer?
```

## Arguments

<value>::<NRf> specifies the filter bandwidth resolution in the Offset & Limit table.

## Returns

<NR3>

## Examples

SENSe:LTE:ACLR:OFFSet1:BANDwidth:RESolution:FILTer? might return 100.000000000E+3 indicating the value of filter bandwidth of the first row is 100 kHz.

## SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo

Sets or queries the resolution video bandwidth (VBW) value in the specified row in the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo <value>
```

```
SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo?
```

## Arguments

<value>::<NRf> specifies the VBW in the specifies row.

## Returns

<NR3>

## Examples

`SENSe:LTE:ACLR:OFFSet1:BANDWIDTH:RESOLUTION:VIDEO 40.00000000E+3` sets the resolution VBW to 40 kHz for row 1.

## **SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo:STATe**

Sets or queries whether to enable or disable the video resolution bandwidth (VBW) in the specified row in the Offsets & Limits table for the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

### Group

Sense commands

### Syntax

`SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo:STATe{ 0 | 1 }`

`SENSe:LTE:ACLR:OFFSet<x>:BANDwidth:RESolution:VIDeo:STATe?`

### Arguments

0 disables the VBW.

1 enables the VBW.

### Returns

0 means the resolution VBW is disabled in the specified row.

1 means the VBW is enabled in the specified row.

## Examples

`SENSe:LTE:ACLR:OFFSet1:BANDWIDTH:RESOLUTION:VIDEO:STATE 1` enables the resolution VBW for row 1.

## **SENSe:LTE:ACLR:OFFSet<x>:DETection**

Sets or queries the detection settings for all offsets for the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<x>:DETEction { AVERAge | PEAK }
SENSe:LTE:ACLR:OFFSet<x>:DETEction?
```

## Arguments

AVERAge: sets the offsets setting Average (VRMS)

PEAK: sets the offsets setting is the positive peak

## Returns

AVERAge means the offsets setting Average (VRMS)

PEAK means the offsets setting is the positive peak

## Examples

```
SENSe:LTE:ACLR:OFFSet<x>:DETECTION AVER sets the detection settings for all offsets to Average RMS.
```

## SENSe:LTE:ACLR:OFFSet<x>:FREQuency:BANDwidth

Sets or queries the frequency bandwidth value for the specified row of the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:BANDwidth <value>
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:BANDwidth?
```

## Arguments

<value>::<NRf> specifies the frequency bandwidth value

## Returns

<NR3>

## Examples

```
SENSe:LTE:ACLR:OFFSet1:FREQUENCY:BANDWIDTH? might return 4.515000000E+6, indicating frequency bandwidth value of row 1 in the Offset & Limits table.
```

## **SENSe:LTE:ACLR:OFFSet<x>:FREQuency:OFFSet**

Sets or queries the frequency offset value of the specified row in the Offsets & Limits table for the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

### **Group**

Sense commands

### **Syntax**

```
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:OFFSet <value>
```

```
SENSe:LTE:ACLR:OFFSet<x>:FREQuency:OFFSet?
```

### **Arguments**

<value>::<NRf> specifies the frequency offset value

### **Returns**

<NR3>

### **Examples**

SENSe:LTE:ACLR:OFFSet1:FREQuency:OFFSet? might return 5.000000000E+6 indicating frequency offset value of row 1 in the Offset & Limits table tab.

## **SENSe:LTE:ACLR:OFFSet<x>:LIMit:ABSolute**

Sets or queries the absolute limit value of the row in the Offset & Limit table tab for the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

### **Group**

Sense commands

### **Syntax**

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:ABSolute <value>
```

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:ABSolute?
```

### **Arguments**

<value>::<NRf> specifies the absolute limit value of the specified row in the Offset & Limit table.

## Returns

<NRf>

## Examples

`SENSe:LTE:ACLR:OFFSet1:LIMit:ABSOLUTE?` might return `-6.450000000` indicating the absolute limit value of row 1 is `-6.450000000` dBm.

## **SENSe:LTE:ACLR:OFFSet<x>:LIMit:MASK**

Sets or queries the limit mask setting for the specified row in the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

Parameter <x> represents the row in the Offsets & Limits table

## Group

Sense commands

## Syntax

`SENSe:LTE:ACLR:OFFSet<x>:LIMit:MASK { ABS | REL | AND | OR }`

`SENSe:LTE:ACLR:OFFSet<x>:LIMit:MASK?`

## Arguments

ABS specifies the limit mask is set to Absolute.

REL specifies the limit mask is set to Relative

AND specifies the limit mask is set to Absolute AND Relative.

OR specifies the limit mask is set to Absolute OR Relative.

## Returns

ABS means that the limit mask is set to Absolute.

REL means that the limit mask is set to Relative

AND means that the limit mask is set to Absolute AND Relative.

OR means that the limit mask is set to Absolute OR Relative.

## Examples

`SENSe:LTE:ACLR:OFFSet3:LIMit:MASK REL` sets the limit mask for row 3 to Relative.

## **SENSe:LTE:ACLR:OFFSet<x>:LIMit:RELative**

Sets or queries the relative limit value of channel under Offset & Limit table tab for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

Parameter <x> represents the row in the Offsets & Limits table

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:RELative <value>
```

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:RELative?
```

## Arguments

<value>::<Nrf> specifies the relative limit value of the specified row in the Offset & Limit table.

## Returns

<Nrf>

## Examples

SENSe:LTE:ACLR:OFFSet1:LIMit:RELATIVE? might return -44.2000000000, indicating the relative limit value of row 1 is -44.2000000000 dB.

## SENSe:LTE:ACLR:OFFSet<x>:LIMit:SIDE

Sets or queries the side of the offset limit for the specified row in the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

Parameter <x> represents the row in the Offsets & Limits table

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:SIDE { LEFT | RIGHT | BOTH }
```

```
SENSe:LTE:ACLR:OFFSet<x>:LIMit:SIDE?
```

## Arguments

LEFT specifies the left offset limit.

RIGHT specifies the right offset limit.

BOTH specifies both sides of the offset limit.



## Returns

LEFT means the left side offset limit is used for the specified row. RIGHT means the right side offset limit is used for the specified row. BOTH means both sides of the offset limit are used for the specified row.

## Examples

`SENSe:LTE:ACLR:OFFSET1:LIMIT:SIDE BOTH` specifies the measurement uses both sides of the offset in row 1.

## SENSe:LTE:ACLR:OFFSet<x>:STATe

Sets or queries the specified row of the Offsets & Limits table for the LTE ACLR display.

## Conditions

Measurement view: LTE ACLR

The parameter <x> represents the row in the Offsets & Limits table.

## Group

Sense commands

## Syntax

`SENSe:LTE:ACLR:OFFSet<x>:STATe { OFF | ON | 0 | 1 }`

`SENSe:LTE:ACLR:OFFSet<x>:STATe?`

## Arguments

OFF or 0 turns off display of the row in the measurement view. ON or 1 turns on display of the row in the measurement view.

## Returns

OFF or 0 means the display of the row in the measurement view is off. ON or 1 means the display of the row in the measurement view is on.

## Examples

`SENSe:LTE:ACLR:OFFSet2:STATe OFF` means that the display of row 2 in the Offsets & Limits table.

## SENSe:LTE:ACLR:POINTs:COUNT

Sets or queries the number of points used for the trace in the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

SENSe:LTE:ACLR:POINTs:COUNT <value>

SENSe:LTE:ACLR:POINTs:COUNT?

## Arguments

<value>::<Nrf> specifies the number of points used for the trace

## Returns

P<NR1>

## Examples

SENSe:LTE:ACLR:POINTS:COUNT? might return P8001, indicating a maximum of 8001 trace points can be used.

## SENSe:LTE:ACLR:REFeRence:MAGNitude

Sets or queries the reference power magnitude for the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

SENSe:LTE:ACLR:REFeRence:MAGNitude <value>

SENSe:LTE:ACLR:REFeRence:MAGNitude?

## Arguments

<value>::<Nrf> specifies the reference power magnitude

## Returns

<Nrf>

## Examples

SENSe:LTE:ACLR:REFERENCE:MAGNITUDE? might return a -51.057, indicating that the reference power magnitude is -51.057 dBm.

## SENSe:LTE:ACLR:REFeRence:MAGNitude:AUTO

Sets or queries the reference magnitude to automatic or manual for the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:REFerence:MAGNitude:AUTO { 0 | 1 }
```

```
SENSe:LTE:ACLR:REFerence:MAGNitude:AUTO?
```

## Arguments

0 sets reference magnitude to Manual.

1 sets reference magnitude to Automatic.

## Returns

0 means the reference magnitude is set to Manual.

1 means the reference magnitude is set to Automatic.

## Examples

```
SENSe:LTE:ACLR:REFERENCE:MAGNITUDE:AUTO 0 sets the reference magnitude to Manual.
```

## SENSe:LTE:ACLR:REFerence:MAGNitude:DETection

Sets or queries the reference magnitude detection for the LTE ACLR measurement.

## Conditions

Measurement view: LTE ACLR

## Group

Sense commands

## Syntax

```
SENSe:LTE:ACLR:REFerence:MAGNitude:DETection { AVERage | PEAK }
```

```
SENSe:LTE:ACLR:REFerence:MAGNitude:DETection?
```

## Arguments

AVER sets reference magnitude detection to Avg(VRMS) PEAK sets reference magnitude detection to +Peak.

## Returns

AVER means the reference magnitude detection is set to Avg(VRMS) PEAK means the reference magnitude detection is set to +Peak.

## Examples

```
SENSe:LTE:ACLR:REFERENCE:MAGNITUDE:DETECTION AVER sets the magnitude detection to Avg(RMS).
```

## **SENSe:LTE:ACLR:STEP:AUTO**

Determines whether to automatically update the center frequency increment/decrement step size in the LTE ACLR display. The step value controls the magnitude of each step when using the front panel controls or arrow key on the keyboard.

### **Conditions**

Measurement view: LTE ACLR

### **Group**

Sense commands

### **Syntax**

```
SENSe:LTE:ACLR:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
SENSe:LTE:ACLR:STEP:AUTO?
```

### **Arguments**

OFF or 0 turns off the automatic update.

ON or 1 turns on automatic updates center frequency.

### **Returns**

OFF or 0 means automatic update is off. ON or 1 means automatic updates are off.

### **Examples**

```
SENSe:LTE:ACLR:STEP:AUTO ON automatically updates the center frequency.
```

## **SENSe:LTE:ACLR:STEP:MAGNitude**

Sets or queries the step magnitude for the center frequency in the LTE ACLR display.

### **Conditions**

Measurement view: LTE ACLR

### **Group**

Sense commands

### **Syntax**

```
SENSe:LTE:ACLR:STEP:MAGNitude <value>
```

```
SENSe:LTE:ACLR:STEP:MAGNitude?
```

### **Arguments**

<value>::<NRf> specifies the center frequency

### **Returns**

<NR3>

## Examples

`SENSe:LTE:ACLR:STEP:MAGNITUDE 4.00000000E+6` sets the step magnitude to 4 MHz.

## SENSe:LTE:ANALysis:AVERage:OVER:OFFSlot:STATe

Determines whether to enable or disable Auto check box of Average over offslot in the Prefs tab of LTE Power vs Time display control panel.

### Conditions

Measurement view: LTE Power vs Time

### Group

Sense commands

### Syntax

```
SENSe:LTE:ANALysis:AVERage:OVER:OFFSlot:STATe {0 | OFF | 1 | ON}
```

```
SENSe:LTE:ANALysis:AVERage:OVER:OFFSlot:STATe?
```

### Arguments

0 or OFF disables the Average over offslot check box 1 or ON enables the Average over offslot check box

### Returns

<Boolean> 0, 1

## Examples

`SENSe:LTE:ANALYSIS:AVERAGE:OVER:OFFSLOT:STATE OFF` disables the Average over offslot check box.

## SENSe]:LTE:ANALysis:EQUALization:STATe

Sets or queries the Enable Equalization state in the Analysis Params tab of the LTE Settings control panel.

### Conditions

Measurement view: LTE Constellation, LTE Power vs Time

### Group

Sense commands

### Syntax

```
SENSe]:LTE:ANALysis:EQUALization:STATe { 0 | 1 }
```

```
SENSe]:LTE:ANALysis:EQUALization:STATe?
```

### Arguments

0 disables the Enable Equalization.

1 enables the Enable Equalization.

### Returns

0 means that Enable Equalization is disabled. 1 means that Enable Equalization is enabled.

### Examples

`SENSe:LTE:ANALYSIS:EQUALIZATION:STATE?` might return 0, indicating that Enable Equalization is disabled.

## **SENSe:LTE:ANALysis:LENGth**

Sets or queries the Analysis length value in the Analysis Time tab of the LTE Power vs Time display and LTE Constellation display control panels.

### Conditions

Measurement view: LTE Power vs Time, LTE Constellation

### Group

Sense commands

### Syntax

`SENSe:LTE:ANALysis:LENGth <Value>`

`SENSe:LTE:ANALysis:LENGth?`

### Arguments

`<Value>=<NR2>`

### Returns

`<NR2>`

### Examples

`SENSe:LTE:ANALYSIS:LENGTH` might return 34.2, indicating the analysis length is 34.2.

## **SENSe:LTE:ANALysis:LENGth:ACTual? (Query Only)**

Queries the actual analysis length for the LTE Constellation and LTE Power vs Time displays. A query returns a value in either seconds or symbols.

### Conditions

Measurement view: LTE Constellation, LTE Power vs Time

### Group

Sense commands

## Syntax

```
SENSe:LTE:ANALysis:LENGth:ACTual?
```

## Returns

<NRf>

## Examples

SENSe:LTE:ANALYSIS:LENGTH:ACTUAL? might return 43, indicating that the actual analysis length is 43 symbols.

## SENSe:LTE:ANALysis:LENGth:AUTO

Determines whether to enable or disable the Auto check box for Analysis Length in the Analysis Time tab of the LTE Power vs Time display and LTE Constellation display control panels.

## Conditions

Measurement view: LTE Power vs Time, LTE Constellation

## Group

Sense commands

## Syntax

```
SENSe:LTE:ANALysis:LENGth:AUTO {0 | OFF | 1 | ON}
```

```
SENSe:LTE:ANALysis:LENGth:AUTO?
```

## Arguments

0 or OFF disables the Auto checkbox of analysis length 1 or ON enables the Auto checkbox of analysis length

## Returns

<Boolean> 0, 1

## Examples

SENSe:LTE:ANALYSIS:LENGTH:AUTO ON enables the Analysis Length Auto check box.

## [SENSe]:LTE:ANALysis:OFFSet

Sets or queries the LTE analysis offset for the LTE Constellation or LTE Power vs Time measurements. The query command returns the result in symbols or seconds, depending on the unit choice selected in the Analysis Time tab of the LTE Settings control panel.

## Conditions

Measurement view: LTE Constellation, LTE Power vs Time

## Group

Sense commands

## Syntax

```
[SENSe]:LTE:ANALysis:OFFSet <value>
```

## Arguments

<value> specifies the analysis offset in the Analysis Time tab of the LTE Settings control panel.

## Returns

<NRf> Units are determined from the units choice selected in the Analysis Time tab of the LTE Settings control panel.

## Examples

```
SENSe:LTE:ANALYSIS:OFFSET -50 sets the analysis offset to -50 symbols.
```

## SENSe:LTE:ANALysis:OFFSet:AUTO

Sets or queries the Auto checkbox state for analysis offset. This setting determines if the analysis offset for the LTE Constellation and LTE Power vs Time displays is set automatically or manually.

## Conditions

Measurement view: LTE Constellation, LTE Power vs Time

## Group

Sense commands

## Syntax

```
SENSe:LTE:ANALysis:OFFSet:AUTO { 0 | 1 }
```

## Arguments

0 sets the analysis offset to manual (box is unchecked)

1 sets the analysis offset to automatic (checkbox is checked)

## Returns

0 means the analysis offset is set to manual (box is unchecked)

1 means the analysis offset is set to automatic (checkbox is checked)

## Examples

```
SENSe:LTE:ANALYSIS:OFFSET:AUTO 1 sets the analysis offset to automatic.
```

## SENSe:LTE:ANALysis:TIME:UNIT

Sets or queries the unit preference under the Analysis tab of the LTE Settings control panel for the LTE Constellation and LTE Power vs Time displays.



## Conditions

Measurement view: LTE Constellation, LTE Power vs Time

## Group

Sense commands

## Syntax

```
SENSe:LTE:ANALySis:TIME:UNIT { SEC | SYMB }
```

```
SENSe:LTE:ANALySis:TIME:UNIT?
```

## Arguments

SEC sets the units to seconds.

SYMB sets the units to symbols.

## Returns

SEC means the units are set to seconds.

SYMB means the units are set to symbols.

## Examples

```
SENSe:LTE:ANALYSIS:TIME:UNIT SEC sets the units to seconds.
```

## SENSe:LTE:ANTEнна:PORTs:CHOSen

Sets or queries the chosen port in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

## Conditions

Measurement view: LTE Power vs Time, LTE Constellation

## Group

Sense commands

## Syntax

```
SENSe:LTE:ANTEнна:PORTs:CHOSen { ZERO | ONE | TWO | THREE }
```

```
SENSe:LTE:ANTEнна:PORTs:CHOSen?
```

## Arguments

{ ZERO | ONE | TWO | THREE }

## Returns

<String> (ZERO to THREE)

## Examples

`SENSe:LTE:ANTENNA:PORTS:CHOSEN ONE` sets the chosen port to port 1.

## SENSe:LTE:ANTENNA:PORTS:TOTAl

Sets or queries the total number of Antenna ports in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

### Conditions

Measurement view: LTE Power vs Time, LTE Constellation

### Group

Sense commands

### Syntax

`SENSe:LTE:ANTENNA:PORTS:TOTAl {ONE|TWO|FOUR}`

`SENSe:LTE:ANTENNA:PORTS:TOTAl?`

### Arguments

{ONE|TWO|FOUR}

### Returns

<String> ONE , TWO or FOUR

## Examples

`SENSe:LTE:ANTENNA:PORTS:TOTAL FOUR` sets the total number of Antenna ports to 4.

## SENSe:LTE:AUTO:DETECT

Determines whether to enable or disable the Auto check box in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panel to automatically detect.

### Conditions

Measurement view: LTE Power vs Time, LTE Constellation

### Group

Sense commands

### Syntax

`SENSe:LTE:AUTO:DETECT {0 | OFF | 1 | ON}`

`SENSe:LTE:AUTO:DETECT?`

## Arguments

ON or 1 to enables the Auto check box OFF or 0 to disables the Auto check box

## Returns

<Boolean> 0, 1

## Examples

SENSe:LTE:AUTO:DETECT 1 enables the Auto detect check box.

## SENSe:LTE:CHANnel:(BANDwidth|BWIDth)

Sets or queries Channel Bandwidth value in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

## Conditions

Measurement view: LTE Power vs Time, LTE Constellation

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHANnel:(BANDwidth|BWIDth) {BW1PT4M|BW3M| BW5M| BW10M| BW15M| BW20M}
SENSe:LTE:CHANnel:(BANDwidth|BWIDth)?
```

## Arguments

BW1PT4M sets the value to 1.4MHz

BW3M sets the value to 3 MHz

BW5M sets the value to 5 MHz

BW10M sets the value to 10 MHz

BW15M sets the value to 15 MHz

BW20M sets the value to 20 MHz

## Returns

<String> See Arguments

## Examples

SENSe:LTE:CHANNEL:(BANDWIDTH|BWIDTH) BW15M sets the Channel Bandwidth to 15 MHz.

## SENSe:LTE:CHANnel:BANDwidth

Specifies or queries the channel bandwidth to use for the LTE Constellation or LTE Power vs Time measurements.

## Conditions

Measurement view: LTE Constellation, LTE Power vs Time

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHANnel:BANDwidth { BW5M | BW10M | BW15M | BW20M }
```

## Arguments

BW5M, BW10M, BW15M, BW20M each specifies a channel bandwidth to use for the measurement.

## Returns

Query returns one of the argument values, indicating the selected channel bandwidth.

## Examples

```
SENSe:LTE:CHANNEL:BANDWIDTH BW5M sets the channel bandwidth to use to 5 MHz.
```

## SENSe:LTE:CHSPectrum:AVERage

Sets or queries how to average waveform in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPectrum:AVERage { OFF | TIME | FREQ }
```

```
SENSe:LTE:CHSPectrum:AVERage?
```

## Arguments

OFF disables averaging.

TIME averages the waveform using time samples.

FREQuency averages the waveform using frequency samples.

## Returns

OFF means averaging is disabled.

TIME means averaging of the waveform is performed using time samples.

FREQuency means averaging of the waveform is performed using frequency samples.

## Examples

`SENSe:LTE:CHSPECTRUM:AVERAGE TIME` sets the analyzer to average the waveform using time samples.

## **SENSe:LTE:CHSPectrum:AVERage:COUNT**

Sets or queries the number of traces for averaging in the LTE Channel Spectrum display.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Sense commands

### Syntax

`SENSe:LTE:CHSPectrum:AVERage:COUNT <number>`

`SENSe:LTE:CHSPectrum:AVERage:COUNT?`

### Arguments

`<number>::<NR1>` specifies the average count

### Returns

`<NR1>`

## Examples

`SENSe:LTE:CHSPECTRUM:AVERAGE:COUNT?` might return 10, indicating that 10 is the number of traces for averaging.

## **SENSe:LTE:CHSPectrum:BANDwidth:RESolution**

Sets or queries the resolution bandwidth (RBW) in the LTE Channel Spectrum display.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Sense commands

### Syntax

`SENSe:LTE:CHSPectrum:BANDwidth:RESolution`

### Arguments

`<value>::<NRf>` specifies the resolution bandwidth

## Returns

<NR3>

## Examples

`SENSe:LTE:CHSPECTRUM:BANDWIDTH:RESOLUTION?` might return `30.000000E+3`, indicating the RBW is 30 kHz.

## **SENSe:LTE:CHSPepectrum:BA NDwidth:RESOlution:AUTO**

Determines whether to set the resolution bandwidth (RBW) in the LTE Channel Spectrum display automatically or manually.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

`SENSe:LTE:CHSPepectrum:BA NDwidth:RESOlution:AUTO { OFF | ON| 0 | 1 }`

`SENSe:LTE:CHSPepectrum:BA NDwidth:RESOlution:AUTO?`

## Arguments

OFF or 0 sets the RBW to manual.

ON or 1 sets the RBW to automatic.

## Returns

OFF or 0 means that the RBW is set to manual.

ON or 1 means that the RBW is set to automatic.

## Examples

`SENSe:LTE:CHSPECTRUM:BANDWIDTH:RESOLUTION:AUTO 1` sets the RBW to automatic and is decided by the analysis module.

## **SENSe:LTE:CHSPepectrum:BA NDwidth:VIDeo**

Sets or queries the video bandwidth (VBW) in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPepectrum:BA NDwidth:VIDeo <value>
```

## Arguments

<value>::<NRf> specifies the video bandwidth

## Returns

<NR3>

## Examples

```
SENSe:LTE:CHSPECTRUM: BANDWIDTH: VIDEO 3.000000E+3 sets the VBW to 3 kHz.
```

## SENSe:LTE:CHSPepectrum:BA NDwidth:VIDeo:STATe

Sets or queries enabling or disabling the video bandwidth (VBW) in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPECTRUM: BANDWIDTH: VIDEO: STATe { 0 | 1 }
```

```
SENSe:LTE:CHSPECTRUM: BANDWIDTH: VIDEO: STATe?
```

## Arguments

0 disables the VBW.

1 enables the VBW.

## Returns

0 means that the VBW is disabled.

1 means that the VBW is enabled.

## Examples

```
SENSe:LTE:CHSPECTRUM: BANDWIDTH: VIDEO: STATe 1 sets the VBW to enabled.
```

## SENSe:LTE:CHSPepectrum:CHANnel:BA NDwidth

Specifies or queries a nominal channel bandwidth to use for the LTE Channel Spectrum measurements.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPepectrum:CHANnel:BANDwidth { BW5M | BW10M | BW20M | BW1PT4M | BW3M | BW15M }
```

## Arguments

BW5M, BW10M, BW20M, BW1PT4M, BW3M, BW15M each specifies a nominal channel bandwidth to use for the measurement.

## Returns

Query returns one of the argument values, indicating the selected nominal channel bandwidth.

## Examples

```
SENSe:LTE:CHSPepectrum:CHANNEL:BANDWIDTH BW20M sets the nominal channel bandwidth to use to 20 MHz.
```

## SENSe:LTE:CHSPepectrum:CLEar:RESults (No Query Form)

Clears the results in the LTE Channel Spectrum measurement.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPepectrum:CLEar:RESults
```

## Arguments

None

## Examples

```
SENSe:LTE:CHSPepectrum:CLEAR:RESULTS clears the results of the LTE Channel Spectrum measurement.
```

## SENSe:LTE:CHSPepectrum:DETection

Sets or queries the trace detection settings for the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum



## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPectrum:DETection { POS | AVER }
```

```
SENSe:LTE:CHSPectrum:DETection?
```

## Related Commands

## Arguments

AVERAge: sets the detection setting to Average (VRMS) POS: sets the detection setting to the positive peak

## Returns

AVERAge means the detection setting is Average (VRMS) POS means the detection setting is positive peak

## Examples

SENSe:LTE:CHSPECTRUM:DETECTION POS sets the trace detection settings to +Peak under the Prefs tab.

## SENSe:LTE:CHSPectrum:FREQuency

Sets or queries the measurement center frequency in the LTE Channel Spectrum display.

## Conditions

Measurement view: LTE Channel Spectrum

## Group

Sense commands

## Syntax

```
SENSe:LTE:CHSPectrum:FREQuency <value>
```

```
SENSe:LTE:CHSPectrum:FREQuency?
```

## Arguments

<value>::<Nrf> specifies the center frequency

## Returns

<NR3>

## Examples

SENSe:LTE:CHSPECTRUM:FREQUENCY 1.00000000E+9 sets the center frequency to 1.0 GHz.

## **SENSe:LTE:CHSPectrum:FREQuency:SPAN**

Sets or queries the frequency span in the LTE Channel Spectrum measurement.

### **Conditions**

Measurement view: LTE Channel Spectrum

### **Group**

Sense commands

### **Syntax**

`SENSe:LTE:CHSPectrum:FREQuency:SPAN <value>`

`SENSe:LTE:CHSPectrum:FREQuency:SPAN?`

### **Arguments**

`<value>::<NRf>` specifies the frequency span

### **Returns**

`<NRf>`

### **Examples**

`SENSe:LTE:CHSPECTRUM:FREQUENCY:SPAN` sets the frequency span in the LTE Channel Spectrum display.

## **SENSe:LTE:CHSPectrum:FREQuency:STEP**

Sets or queries the frequency step size in the LTE Channel Spectrum measurement. Programming a specified step size sets

`SENSe:LTE:CHSPectrum:FREQuency:STEP:AUTO` to `OFF`.

### **Conditions**

Measurement view: LTE Channel Spectrum

### **Group**

Sense commands

### **Syntax**

`SENSe:LTE:CHSPectrum:FREQuency:STEP <value>`

`SENSe:LTE:CHSPectrum:FREQuency:STEP?`

### **Arguments**

`<value>::<NRf>` specifies the frequency step size

### **Returns**

`<NR3>`

## Examples

`SENSe:LTE:CHSPECTRUM:FREQUENCY:STEP 200.00000000E+3` sets the frequency step size to 200 kHz.

## SENSe:LTE:CHSPectrum:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the LTE Channel Spectrum measurement.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Sense commands

### Syntax

```
SENSe:LTE:CHSPectrum:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
SENSe:LTE:CHSPectrum:FREQuency:STEP:AUTO?
```

### Arguments

OFF or 0 sets the frequency step size is set to manual. ON or 1 sets the frequency step size is set to automatic.

### Returns

OFF or 0 means that the frequency step size is set to manual. ON or 1 means that the frequency step size is set to automatic.

## Examples

`SENSe:LTE:CHSPECTRUM:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.

## SENSe:LTE:CHSPectrum:POINts:COUNT

Sets or queries the number of points used for the trace in the LTE Channel Spectrum measurement.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Sense commands

### Syntax

```
SENSe:LTE:CHSPectrum:POINts:COUNT <value>
```

```
SENSe:LTE:CHSPectrum:POINts:COUNT?
```

### Arguments

`<value>::<Nrf>` specifies the number of points used for the trace

## Returns

P<NR1>

## Examples

`SENSE:LTE:CHSPECTRUM:POINTS:COUNT?` might return P8001, indicating a maximum of 8001 trace points can be used.

## SENSE:LTE:CYCLic:PREFix

Sets or queries the cyclic prefix type in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

## Conditions

Measurement view: LTE Power vs Time, LTE Constellation

## Group

Sense commands

## Syntax

`SENSE:LTE:CYCLic:PREFix {NORM|EXT}`

`SENSE:LTE:CYCLic:PREFix?`

## Arguments

`NORM` sets the cyclic prefix to Normal.

`EXT` sets the cyclic prefix to Extended.

## Returns

<String> NORM or EXT

## Examples

`SENSE:LTE:CYCLIC:PREFIX NORM` sets the cyclic prefix to Normal.

## SENSE:LTE:FRAME:STRUcture

Sets or queries the frame structure in the Modulation Params tab of the LTE settings control panel.

## Conditions

Measurement view: LTE Constellation and LTE Power vs Time

## Group

Sense commands

## Syntax

`SENSE:LTE:FRAME:STRUcture { FDD | TDD }`

SENSe:LTE:FRAME:STRUCTure?

### Arguments

FDD  
TDD

### Returns

FDD TDD

### Examples

SENSe:LTE:FRAME:STRUCTURE FDD sets the frame structure to FDD.

## SENSe:LTE:LINK:CONFIguration

Sets or queries Uplink-Downlink value in TDD configuration in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

This happens only when Downlink TDD is Selected as Frame Structure

### Conditions

Measurement view: LTE Power vs Time, LTE Constellation

### Group

Sense commands

### Syntax

SENSe:LTE:LINK:CONFIguration {ZERO |ONE |TWO |THREE |FOUR|FIVE |SIX}  
SENSe:LTE:LINK:CONFIguration?

### Arguments

{ZERO |ONE |TWO |THREE |FOUR |FIVE |SIX}

### Returns

<String> (Zero to Six)

### Examples

SENSe:LTE:LINK:CONFIgURATION ONE sets the Uplink-Downlink value as 1 in TDD configuration.

## SENSe:LTE:SUBFrame:CONFIguration

Sets or queries special subframe value in TDD configuration in the Modulation Params tab of the LTE Power vs Time display and LTE Constellation display control panels.

This happens only when Downlink TDD is selected as the Frame Structure

## Conditions

Measurement view: LTE Power vs Time, LTE Constellation

## Group

Sense commands

## Syntax

```
SENSe:LTE:SUBFrame:CONFiguration {ZERO |ONE |TWO |THREE|FOUR |FIVE |SIX |SEVEN |EIGHT |NINE}
```

```
SENSe:LTE:SUBFrame:CONFiguration?
```

## Arguments

```
{ZERO |ONE |TWO |THREE |FOUR |FIVE |SIX |SEVEN |EIGHT |NINE}
```

## Returns

<String> (Zero to Nine)

## Examples

SENSe:LTE:SUBFRAME:CONFIGURATION SIX sets the special subframe value to 6 in TDD configuration.

## [SENSe]:MCPower:AVERage

Sets or queries the average method in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:AVERage { OFF | TIME | FREQuency }
```

```
[SENSe]:MCPower:AVERage?
```

## Arguments

OFF disables averaging.

TIME performs averaging for time samples.

FREQuency performs averaging for frequency samples.

## Examples

SENSe:MCPOWER:AVERAGE TIME performs averaging for time samples.

## [SENSe]:MCPower:AVERage:COUNT

Sets or queries the average count in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Sense commands

### Syntax

```
[SENSe]:MCPower:AVERage:COUNT <value>
```

```
[SENSe]:MCPower:AVERage:COUNT?
```

### Arguments

<value>::<NR1> specifies the average count. Range: 2 to 10000.

### Examples

SENSe:MCPOWER:AVERAGE:COUNT 256 sets the average count to 256.

## [SENSe]:MCPower:{BANDwidth|BWIDth}{:RESolution}

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:MCPower{BANDwidth|BWIDth}{:RESolution}:AUTO OFF.

### Conditions

Measurement views: MCPR

### Group

Sense commands

### Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}{:RESolution} <value>
```

```
[SENSe]:MCPower:{BANDwidth|BWIDth}{:RESolution}?
```

### Related Commands

```
[SENSe]:MCPower:{BANDwidth|BWIDth}{:RESolution}:AUTO
```

### Arguments

<value>::<NRf> specifies the RBW. Range: 100 Hz to 5 MHz.

### Examples

SENSe:MCPOWER:BANDWIDTH:RESOLUTION 200kHz sets the RBW to 200 kHz.

## **[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the MCPR measurement.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]:ACTual?
```

### **Arguments**

None

### **Returns**

<NRf> The actual RBW in Hz.

### **Examples**

`SENSe:MCPOWER:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

## **[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]:AUTO**

Determines whether to set the resolution bandwidth (RBW) automatically or manually.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]:AUTO?
```

### **Arguments**

OFF or 0 specifies that the resolution bandwidth is set manually using the `[SENSe]:MCPower:{BANDwidth|BWIDth}[ :RESolution]` command.

ON or 1 specifies that the resolution bandwidth is set automatically.



## Examples

`SENSe:MCPOWER:BANDWIDTH:AUTO ON` sets the resolution bandwidth automatically.

## **[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets `[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe OFF`.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo <value>
```

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo?
```

## Related Commands

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe
```

## Arguments

`<value>::<Nrf>` specifies the VBW.

Range: Current RBW/104 (1 Hz minimum) to Current RBW.

## Examples

`SENSe:MCPOWER:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

## **[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe**

Determines whether to enable the video bandwidth (VBW) in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:MCPower:{BANDwidth|BWIDth}:VIDeo:STATe?
```

## Arguments

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

## Examples

`SENSe:MCPOWER:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

## [SENSe]:MCPower:CHANnel:ADJacent:ADD (No Query Form)

Adds a pair of upper and lower adjacent channels in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

`[SENSe]:MCPower:CHANnel:ADJacent:ADD <offset>,<bandwidth>`

## Arguments

`<offset>::<NRf>` specifies the offset from the center frequency for the adjacent channel.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

`<bandwidth>::<NRf>` specifies the bandwidth of the adjacent channel.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:MCPOWER:CHANNEL:ADJACENT:ADD 200kHz, 80kHz` adds a pair of upper and lower adjacent channels with the offset of  $\pm 200$  kHz and the bandwidth of 80 kHz.

## [SENSe]:MCPower:CHANnel:ADJacent:DELeTe (No Query Form)

Deletes a selected adjacent channel in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:CHANnel:ADJacent:DELeTe <channel>
```

## Arguments

<channel>::<string> specifies the channel to be deleted.

Specify the channel with "A<n>" for the adjacent channel where <n> represents the channel number (<n> = 1, 2, 3,...). See the example below.

## Examples

```
SENSe:MCPower:CHANnel:ADJacent:DELeTe "A2" deletes A2 (the adjacent channel 2).
```

## [SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>

This command returns the selected adjacent upper and lower limit values. Selection of adjacent channel.

## Conditions

Measurement view: P25 MCPR

The parameter <x>=1 to 100.

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>
```

```
[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>?
```

## Arguments

<x>=1 to 100

## Returns

Upper and lower limit values for selected adjacent channels.

## Examples

```
SENSe:MCPower:CHANnel:ADJacent:LIMIts2? returns upper and lower limit values of A2 (Adjacent channel 2).
```

## **[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>:ENABLE? (Query Only)**

This query returns whether the adjacent channel is selected or not.

### **Conditions**

Measurement view: P25 MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:CHANnel:ADJacent:LIMIts<x>:ENABLE?
```

### **Returns**

0 means the adjacent channel is disabled.

1 means the adjacent channel is enabled.

### **Examples**

SENSe:MCPower:CHANnel:ADJacent:LIMIts1:ENABLE? 1 means that A1 (adjacent channel 1) is enabled.

## **[SENSe]:MCPower:CHANnel:FILTer**

Sets or queries the adjacent channel filter in the MCPR measurement.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:CHANnel:FILTer { RRCosine | NONE }
```

```
[SENSe]:MCPower:CHANnel:FILTer?
```

### **Arguments**

RRCosine selects the Root-Raised-Cosine filter.

NONE uses no filter.

### **Examples**

SENSe:MCPOWER:CHANNEL:FILTER RRCosine selects Root-Raised-Cosine for the adjacent channel filter.

## **[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}**

Sets or queries the frequency bandwidth of the main channels (all share the same value) in the MCPR measurement.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth} <value>
```

```
[SENSe]:MCPower:CHANnel:MAIN:{BANDwidth|BWIDth}?
```

### **Arguments**

<value>::<Nrf> specifies the main channel bandwidth.

Range: 1 Hz to full span.

### **Examples**

SENSe:MCPOWER:CHANNEL:MAIN:BANDWIDTH 4.5MHz sets the main channel bandwidth to 4.5 MHz.

## **[SENSe]:MCPower:CHANnel:MAIN:COUNT**

Sets or queries the number of main channels in the MCPR measurement. You can use this command to add and remove main channels.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:CHANnel:MAIN:COUNT <value>
```

```
[SENSe]:MCPower:CHANnel:MAIN:COUNT?
```

### **Arguments**

<value>::<Nrf> specifies the number of main channels. Range: 1 to 99.

### **Examples**

SENSe:MCPOWER:CHANNEL:MAIN:COUNT 3 sets the the number of main channels to 3.

## [SENSe]:MCPower:CHANnel:MAIN:INACTive

Makes a specified main channel inactive. You can set it on or off. The query returns all inactive main channels.

### Conditions

Measurement views: MCPR

### Group

Sense commands

### Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:INACTive <channel>,<boolean>
```

```
[SENSe]:MCPower:CHANnel:MAIN:INACTive?
```

### Arguments

<channel>::<string> specifies the channel to be inactive.

Specify the channel with "M<n>" for the main channel where <n> represents the channel number (<n> = 1, 2, 3,...). See the example below.

<boolean>::={ OFF | ON | 0 | 1 } specifies that the specified channel is inactive (On) or not (Off).

### Examples

```
SENSe:MCPOWER:CHANNEL:MAIN:INACTIVE "M2",ON makes the main channel 2 inactive.
```

## [SENSe]:MCPower:CHANnel:MAIN:SPACing

Sets or queries frequency difference between centers of each main channel in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Sense commands

### Syntax

```
[SENSe]:MCPower:CHANnel:MAIN:SPACing <value>
```

```
[SENSe]:MCPower:CHANnel:MAIN:SPACing?
```

### Arguments

<value>::<NRf> specifies the spacing between two adjacent main channels.

Range: 1 Hz to 1 GHz.

## Examples

`SENSe:MCPOWER:CHANNEL:MAIN:SPACING 5MHz` sets the main channel spacing to 5 MHz.

## [SENSe]:MCPower:CHIPrate

Sets or queries the chip rate in the MCPR measurement. This command is valid when `[SENSe]:MCPower:CHANnel:FILTer` is set to RRCosine (Root-Raised-Cosine).

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

`[SENSe]:MCPower:CHIPrate <value>`

`[SENSe]:MCPower:CHIPrate?`

## Arguments

`<value>:::<Nrf>` specifies the chip rate. Range: 100 Hz to 105 MHz.

## Examples

`SENSe:MCPOWER:CHIPRATE 1kHz` sets the chip rate to 1 kHz.

## [SENSe]:MCPower:CLEar:RESults (No Query Form)

Restarts the average trace.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

`[SENSe]:MCPower:CLEar:RESults`

## Arguments

None

## Examples

`SENSe:MCPOWER:CLEAR:RESULTS` restarts the average trace.

## **[SENSe]:MCPower:FREQuency**

Sets or queries the center frequency in the MCPR measurement.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:FREQuency <value>
```

```
[SENSe]:MCPower:FREQuency?
```

### **Arguments**

<value>: : <NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

### **Examples**

SENSe:MCPOWER:FREQUENCY 2.35GHz sets the center frequency to 2.35 GHz.

## **[SENSe]:MCPower:FREQuency:STEP**

Sets or queries the frequency step size. Programming a specified step size sets [SENSe]:MCPower:FREQuency:STEP:AUTO OFF.

### **Conditions**

Measurement views: MCPR

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:MCPower:FREQuency:STEP <value>
```

```
[SENSe]:MCPower:FREQuency:STEP?
```

### **Related Commands**

```
[SENSe]:MCPower:FREQuency:STEP:AUTO
```



## Arguments

<value> : : <NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106AB – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:MCPOWER:FREQUENCY:STEP 50kHz` sets the frequency step size to 50 kHz.

## [SENSe]:MCPower:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the MCPR measurement.

### Conditions

Measurement views: MCPR

### Group

Sense commands

### Syntax

```
[SENSe]:MCPower:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:MCPower:FREQuency:STEP:AUTO?
```

### Arguments

OFF or 0 specifies that the frequency step size is set manually using the `[SENSe]:MCPower:FREQuency:STEP` command.

ON or 1 specifies that the frequency step size is set automatically.

### Examples

`SENSe:MCPOWER:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.

## [SENSe]:MCPower:NFLoor:STATe

Determines whether to enable correction for noise floor.

### Conditions

Measurement views: MCPR

### Group

Sense commands

## Syntax

```
[SENSe]:MCPower:NFLoor:STATe { OFF | ON | 0 | 1 }  
[SENSe]:MCPower:NFLoor:STATe?
```

## Arguments

OFF or 0 disables correction for noise floor.

ON or 1 enables correction for noise floor.

## Examples

SENSe:MCPOWER:NFLOOR:STATE ON enables correction for noise floor.

## [SENSe]:MCPower:POINTs:COUNT

Sets or queries the number of trace points in the MCPR display.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:POINTs:COUNT { P801 | P2401 | P4001 | P8001 | P10801 }  
[SENSe]:MCPower:POINTs:COUNT?
```

## Arguments

P801 | P2401 | P4001 | P8001 | P10801 set the number of trace points.

## Examples

SENSe:MCPOWER:POINTS:COUNT P4001 sets the number of trace points in the MCPR measurement to 4001.

## [SENSe]:MCPower:RCHannels? (Query Only)

Queries the power reference in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:RCHannels?
```

## Arguments

None

## Returns

`<power_ref> ::= { Total | M<x> }`

where `<x>` = 1 to 99.

Total indicates that the power reference is the total power of all the active channels.

M<x> indicates that the power reference is the main channel with the index (`<x>`).

## Examples

`SENSe:MCPOWER:RCHANNELS?` might return M3, indicating that the power reference is the main channel 3.

## **[SENSe]:MCPower:RChannels:MAIN<x> (No Query Form)**

Sets the power reference to the main channel with the index (`<x>`) in the MCPR measurement.

The parameter `<x>` = 1 to 99, representing the main channel 1 to 99, respectively. The main channel must be defined using the `s` commands.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

`[SENSe]:MCPower:RChannels:MAIN<x>`

## Related Commands

`[ :SENSe ]:MCPower:CHANnel:MAIN` commands

## Arguments

None

## Examples

`SENSe:MCPOWER:RCHANNELS:MAIN3` selects Main 3 for the power reference channel.

## **[SENSe]:MCPower:RChannels:TOTal (No Query Form)**

Sets the power reference to the total power of all the active channels in the MCPR measurement.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:RCHannels:TOTal
```

## Arguments

None

## Examples

`SENSe:MCPOWER:RCHANNELS:TOTAL` sets the power reference to the total power of all the active channels.

## [SENSe]:MCPower:RRCRolloff

Sets or queries the filter parameter (roll-off ratio) for the Root Raised Cosine filter.

## Conditions

Measurement views: MCPR

## Group

Sense commands

## Syntax

```
[SENSe]:MCPower:RRCRolloff <value>
```

```
[SENSe]:MCPower:RRCRolloff?
```

## Related Commands

```
[SENSe]:MCPower:CHANnel:FILTer
```

## Arguments

`<value> : : <Nrf>` specifies the filter parameter.

Range: 0.001 to 1, 0.0001 step.

## Examples

`SENSe:MCPOWER:RRCROLLOFF 0.3` sets the filter parameter to 0.3.

## [SENSe]:MEASurement:FREQUency[:SOURce<z>]

Sets or queries the measurement frequency.

The parameter `<z>` = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:MEASurement:FREQuency:SOURCE2<value>
```

```
[SENSe]:MEASurement:FREQuency:SOURCE2?
```

## Arguments

<value>: : <Nrf> specifies the measurement frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:MEASurement:FREQuency:SOURCE2? 7.5GHz` sets the measurement frequency frequency of Source 2 to 7.5 GHz.

## [SENSe]:MEASurement:FREQuency[:SOURce<z>]:CENTER:LOCK

Sets or queries locking the measurement center frequency.

The parameter <z> = 1 to number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

Default source is Source 1.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:MEASurement:FREQuency:SOURCE1:CENTER:LOCK { OFF | ON | 0 | 1 }
```

```
[SENSe]:MEASurement:FREQuency:SOURCE1:CENTER:LOCK?
```

## Arguments

OFF or 0 disables locking the center frequency.

ON or 1 enables locking the center frequency.

## Examples

`SENSe:MEASurement:FREQuency:SOURCE1:CENTer:LOCK ON` locks the center frequency.

## [SENSe]:MERRor[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Magnitude error versus Time for a selected Measurement view number

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

[SENSe]:MERRor[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}

[SENSe]:MERRor[:MEASview<y>]:SELEcted:SOURce?

### Arguments

The parameter <y>=1 to 8 represents the measurement index based on the number of sources of connected instruments, where the MEASview keyword is optional by default and y is 1

### Example

`SENSe:MERRor:MEASview1:SELEcted:SOURce?` queries the Source for Magnitude error versus Time Measurement View 1.

## [SENSe]:NOISe:AVERage:COUNT

This command returns the acquisition counts over which the noise average is computed.

### Conditions

Measurement view: Noise Figure

### Group

Sense commands

### Syntax

[SENSe]:NOISe:AVERage:COUNT <integer>

[SENSe]:NOISe:AVERage:COUNT?

## Arguments

<integer>

Range: 1 to 1000

## Returns

<NR3> = a floating point value with an exponent

## Examples

`SENSe:NOISE:AVERAGE:COUNT?` might return `1.000000000E+3`, indicating the average count of the analyzer is set to 1000.

## [SENSe]:NOISe:BANDwidth

This command sets or queries the resolution bandwidth (RBW) for the Noise Figure display measurement.

## Conditions

Measurement view: Noise Figure

## Group

Sense commands

## Syntax

`[SENSe]:NOISe:BANDwidth <NR3>`

`[SENSe]:NOISe:BANDwidth?`

## Arguments

<NR3> = a floating point value with an exponent Range: 50 Hz - 10 MHz

## Returns

<NR3> = a floating point value with an exponent

## Examples

`SENSe:NOISE:BANDWIDTH?` might return `1.000000000E+6`, indicating that the RBW is 1.00 MHz.

## [SENSE]:NOISe:CONTRol:SETTling:OFF[:TIME]

This command sets or queries the analyzer off settle time for the Noise Figure display measurement.

## Conditions

Measurement view: Noise Figure

## Group

Sense commands

## Syntax

```
[SENSE]:NOISE:CONTROL:SETTLING:OFF[:TIME] <NR3>  
[SENSE]:NOISE:CONTROL:SETTLING:OFF[:TIME]?
```

## Related Commands

```
[SENSE]:NOISE:CONTROL:SETTLING:ON[:TIME]
```

## Arguments

<NR3> = a floating point value with an exponent Range: 0 s - 10 ks

## Returns

<NR3> = a floating point value with an exponent

## Examples

SENSE:NOISE:CONTROL:SETTLING:OFF:TIME? might return 200.000000000E-3, indicating the the RSA settle time (off) is 200 ms.

## [SENSE]:NOISE:CONTROL:SETTLING:ON[:TIME]

This command sets or queries the analyzer on settle time for the Noise and Gain display measurements.

## Conditions

Measurement views: Noise Figure, Noise Temperature, Y Factor

## Group

Sense commands

## Syntax

```
[SENSE]:NOISE:CONTROL:SETTLING:ON[:TIME] <NR3>  
[SENSE]:NOISE:CONTROL:SETTLING:ON[:TIME]?
```

## Related Commands

```
[SENSE]:NOISE:CONTROL:SETTLING:OFF[:TIME]
```

## Arguments

<NR3> = a floating point value with an exponent Range: 0 s - 10 ks

## Returns

<NR3> = a floating point value with an exponent

## Examples

SENSE:NOISE:CONTROL:SETTLING:ON:TIME? might return 50.000000000E-3, indicating the RSA settle time (on) is 50 ms.



## [SENSe]:NOISe:CORRection:ENR:CONSTant

This command sets and queries the excess noise ratio (ENR) constant for the Noise Figure and Gain displays.

### Conditions

Measurement view: Noise Figure, Y Factor, Gain, Noise Temperature, Noise Table

### Group

Sense commands

### Syntax

```
[SENSe]:NOISe:CORRection:ENR:CONSTant <NR2>
```

```
[SENSe]:NOISe:CORRection:ENR:CONSTant?
```

### Related Commands

```
[SENSe]:NOISe:CORRection:ENR:MODE
```

### Arguments

<NR2> = a floating point value without an exponent

### Returns

<NR2> = a floating point value without an exponent

### Examples

SENSe:NOISe:CORRection:ENR:CONSTant? might return 10.000000000, indicating that the ENR constant is 10.00 dB.

## [SENSe]:NOISe:CORRection:ENR:MODE

This command sets and queries the Noise Source mode for the Noise Figure and Gain displays.

### Conditions

Measurement view: Noise Figure, Y Factor, Gain, Noise Temperature, Noise Table

### Group

Sense commands

### Syntax

```
[SENSe]:NOISe:CORRection:ENR:MODE { TABLE | CONSTant }
```

```
[SENSe]:NOISe:CORRection:ENR:MODE?
```

### Related Commands

```
[SENSe]:NOISe:CORRection:ENR:CONSTant
```



## Conditions

Measurement view: Noise Figure, Y Factor, Gain, Noise Temperature, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:CORRection:ENR:TABLE:ID <Qstring>
```

```
[SENSe]:NOISe:CORRection:ENR:TABLE:ID?
```

## Related Commands

```
[SENSe]:NOISe:CORRection:ENR:TABLE:DATA?
```

## Arguments

<Qstring> = quoted string of ASCII text

## Returns

<Qstring> = quoted string of ASCII text

## Examples

`SENSe:NOISe:CORRECTION:ENR:TABLE:ID?` might return "346B 10MHz – 18 GHz", indicating the Noise Source model is 346B 10MHz–18 GHz.

## [SENSe]:NOISe:CORRection:ENR:TABLE:NEW (No Query Form)

This command sets the Noise Source Model for the Noise Figure and Gain displays.

## Conditions

Measurement view: Noise Figure, Y Factor, Gain, Noise Temperature, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:CORRection:ENR:TABLE:NEW <Qstring>
```

```
[SENSe]:NOISe:CORRection:ENR:TABLE:NEW?
```

## Related Commands

```
[SENSe]:NOISe:CORRection:ENR:TABLE:DATA?
```

## Arguments

<Qstring> = quoted string of ASCII text

## Returns

<Qstring> = quoted string of ASCII text

## Examples

SENSe:NOISE:CORRECTION:ENR:TABLE:NEW might return "346B 10MHz – 18 GHz", indicating the Noise Source model is 346B 10MHz–18 GHz.

## [SENSe]:NOISe:CORRection:ENR:TABLE:SERIAL

This command sets or queries the Noise Source serial number for the Noise Figure and Gain displays.

## Conditions

Measurement view: Noise Figure, Y Factor, Gain, Noise Temperature, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:CORRection:ENR:TABLE:SERIAL <Qstring>
```

```
[SENSe]:NOISe:CORRection:ENR:TABLE:SERIAL?
```

## Related Commands

```
[SENSe]:NOISe:CORRection:ENR:TABLE:DATA?
```

## Arguments

<Qstring> = quoted string of ASCII text

## Returns

<Qstring> = quoted string of ASCII text

## Examples

SENSe:NOISE:CORRECTION:ENR:TABLE:SERIAL? might return "2037A02334", indicating the Noise Source serial number is 2037A02334.

## [SENSe]:NOISe:CORRection:TCOLd:ADUT:UREFERENCE

Sets or queries the analyzer to use the reference temperature (K) for the loss at the DUT output in the noise displays.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:CORRection:TCOLd:ADUT:UREFERENCE { OFF | ON | 1 | 0 }
[SENSe]:NOISe:CORRection:TCOLd:ADUT:UREFERENCE?
```

## Arguments

OFF or 0 disables the Use Ref temperature feature for the Loss at DUT Output.

ON or 1 enables the Use Ref temperature feature for the Loss at DUT Output.

## Returns

0 means that the Use Ref temperature feature for the Loss at DUT Output is disabled.

1 means that the Use Ref temperature feature for the Loss at DUT Output is enabled.

## Examples

SENSe:NOIS:CORR:TCOL:ADUT:UREF? might return 1, indicating that the temperature for the loss at the DUT output is set to Use Reference temperature.

## [SENSe]:NOISe:CORRection:TCOLd:ADUT[:VALue]

Sets or queries the temperature value (K) for the loss at the DUT output in the noise displays.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:CORRection:TCOLd:ADUT[:VALue] <NR3>
[SENSe]:NOISe:CORRection:TCOLd:ADUT[:VALue]?
```

## Arguments

<NR3> = floating point value with an exponent.

Range: 0 K to 500 K

## Returns

<NR2> = floating point value without an exponent, which is the temperature in Kelvin.

## Examples

SENSe:NOIS:CORR:TCOL:ADUT:VAL? might return 100.000000000, indicating that the temperature for the loss at the DUT output is 100 K.

## **[SENSe]:NOISe:CORRection:TCOLd:BDUT:UREFERENCE**

Sets or queries the analyzer to use the reference temperature (K) for the loss at the DUT input in the noise displays.

### **Conditions**

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, and Noise Table

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NOISe:CORRection:TCOLd:BDUT:UREFERENCE { OFF | ON | 1 | 0 }
```

```
[SENSe]:NOISe:CORRection:TCOLd:BDUT:UREFERENCE?
```

### **Arguments**

OFF or 0 disables the Use Ref temperature feature for the Loss at DUT Input.

ON or 1 enables the Use Ref temperature feature for the Loss at DUT Input.

### **Returns**

0 means that the Use Ref temperature feature for the Loss at DUT Input is disabled.

1 means that the Use Ref temperature feature for the Loss at DUT Input is enabled.

### **Examples**

`SENSe:NOIS:CORR:TCOL:BDUT:UREF?` might return 1, indicating that the temperature for the loss at the DUT input is set to Use Reference temperature.

## **[SENSe]:NOISe:CORRection:TCOLd:BDUT[:VALue]**

Sets or queries the temperature value (K) for the loss at the DUT input in the noise displays.

### **Conditions**

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NOISe:CORRection:TCOLd:BDUT[:VALue] <NR3>
```

```
[SENSe]:NOISe:CORRection:TCOLd:BDUT[:VALue]?
```

### **Arguments**

<NR3> = floating point value with an exponent.

Range: 0 K to 500 K

## Returns

<NR2> = floating point value without an exponent, which is the temperature in Kelvin.

## Examples

`SENSe:NOIS:CORR:TCOL:BDUT:VAL 290` sets the temperature for the loss at the DUT input to 290 K.

## [SENSe]:NOISe:CORRection:TCOLd[:REFeRence]

Sets or queries the reference temperature value (K) in the noise display.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

`[SENSe]:NOISe:CORRection:TCOLd[:REFeRence] <NR3>`

`[SENSe]:NOISe:CORRection:TCOLd[:REFeRence]?`

## Arguments

<NR3> = floating point value with an exponent.

Range: 0 K to 500 K

## Returns

<NR2> = floating point value without an exponent, which is the reference temperature in Kelvin.

## Examples

`SENSe:NOIS:CORR:TCOL:REF 200` sets the reference temperature to 200 K.

## [SENSe]:NOISe:FIGure:POINTs:COUNT (No Query Form)

Sets the number of sample points for the Noise Figure display.

## Conditions

Measurement views: Noise Figure

## Group

Sense commands

## Syntax

`[SENSe]:NOISe:FIGure:POINTs:COUNT { P801 | P2401 | P4001 | P8001 | P10401 }`

## Arguments

P801 sets the number of sample points to 801.

P2401 sets the number of sample points to 2401.

P4001 sets the number of sample points to 4001.

P8001 sets the number of sample points to 8001.

P10401 sets the number of sample points to 10401.

## Examples

`SENSe:NOISE:FIGURE:POINTS:COUNT P801` sets the number of sample points to 801.

## [SENSe]:NOISe:FREQuency:CENTer

Sets or queries the center frequency value (Hz).

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

`[SENSe]:NOISe:FREQuency:CENTer <NRf>`

`[SENSe]:NOISe:FREQuency:CENTer?`

## Arguments

<NRf> is the center frequency value in Hz.

Range: 0 Hz to 6.2 GHz

## Returns

<NR3> = floating point value with an exponent, which is the center frequency value.

## Examples

`SENSe:NOISE:FREQUENCY:CENTER 5.0 GHz` sets the center frequency in the active noise display to 5.0000 GHz.

`SENSe:NOISE:FREQUENCY:CENTER?` might return `6.2000000000E+9`, indicating that the center frequency in the active noise display is 6.2000 GHz.

## [SENSe]:NOISe:FREQuency:MODE

Sets or queries the frequency mode in the noise display.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table



## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:FREQuency:MODE { TABLE | SINGle | SWEpt }
[SENSe]:NOISe:FREQuency:MODE?
```

## Arguments

TABLE: sets the frequency mode to Frequency Table.

SINGle: sets the frequency mode to Single Frequency.

SWEpt: sets the frequency mode to Sweep Frequency.

## Returns

TABL: means the frequency mode is set to Frequency Table.

SING: means the frequency mode is set to Single Frequency.

SWEP: means the frequency mode is set to Sweep Frequency.

## Examples

SENSe:NOIS:FREQ:MODE TABL sets the frequency mode to Frequency Table in the noise display.

## [SENSe]:NOISe:FREQuency:SPAN

Sets or queries the frequency Span value (Hz).

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:FREQuency:SPAN <NRf>
[SENSe]:NOISe:FREQuency:SPAN?
```

## Arguments

<NRf> is the frequency Span value in Hz.

Range: 100 Hz to 6.2 GHz

## Returns

<NR3> = floating point value with an exponent, which is the frequency Span value.

## Examples

`SENSe:NOISE:FREQUENCY:SPAN 3.0 GHz` sets the frequency Span in the active noise display to 3.0000 GHz.

`SENSe:NOISE:FREQUENCY:SPAN?` might return `100.000000000E+3`, indicating that the frequency Span in the active noise display is 100 kHz.

## [SENSe]:NOISe:FREQuency:START

Sets or queries the Start Frequency value.

### Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

### Group

Sense commands

### Syntax

`[SENSe]:NOISe:FREQuency:START <NRf>`

`[SENSe]:NOISe:FREQuency:START?`

### Arguments

`<NRf>` is the Start Frequency value.

### Returns

`<NR3>` = floating point value with an exponent, which is the Start Frequency value.

## Examples

`SENSe:NOISE:FREQUENCY:START 1.5 GHz` sets the Start Frequency in the active noise display to 1.5000 GHz.

`SENSe:NOISE:FREQUENCY:START?` might return `1.500000000E+9`, indicating that the Start Frequency in the active noise display is 1.5000 GHz.

## [SENSe]:NOISe:FREQuency:STEPS

Sets or queries the frequency Steps value.

### Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

### Group

Sense commands

### Syntax

`[SENSe]:NOISe:FREQuency:STEPS <NRf>`

`[SENSe]:NOISe:FREQuency:STEPS?`

## Arguments

<NRf> is the Steps value.

Range: 2 to 999

## Returns

<NRf> is the Steps value.

## Examples

`SENSe:NOISE:FREQUENCY:STEPS 11` sets the frequency Steps value in the active noise display to 11.

## [SENSe]:NOISe:FREQuency:STOP

Sets or queries the Stop Frequency value.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

`[SENSe]:NOISe:FREQuency:STOP <NRf>`

`[SENSe]:NOISe:FREQuency:STOP?`

## Arguments

<NRf> is the Stop Frequency value.

## Returns

<NR3> = floating point value with an exponent, which is the Stop Frequency value.

## Examples

`SENSe:NOISE:FREQUENCY:STOP 1.5 GHz` sets the Stop Frequency in the active noise display to 1.5000 GHz.

`SENSe:NOISE:FREQUENCY:STOP?` might return `1.5000000000E+9`, indicating that the Stop Frequency in the active noise display is 1.5000 GHz.

## [SENSe]:NOISe:FREQuency:TABLE:DATA? (Query Only)

Queries the data in the Frequency Table for the noise display.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:FREQuency:TABLE:DATA?
```

## Returns

<string> which contains each frequency value (Hz) in the frequency table for the noise display.

## Examples

SENSe:NOISE:FREQUENCY:TABLE:DATA? might return "1500000000, 1505000005, 1510000010, 1515000015, 1520000020, 1525000025, 1530000030, 1545000045, 1550000050," indicating each frequency in the Frequency Table.

## [SENSe]:NOISe:FREQuency:TABLE:NEW (No Query Form)

Creates a new Frequency Table with the specified values for the noise display.

## Conditions

Measurement views: Noise Figure, Gain, Noise Temperature, Y Factor, Noise Table

## Group

Sense commands

## Syntax

```
[SENSe]:NOISe:FREQuency:TABLE:NEW <string>
```

## Arguments

<string> = "<NR3>,<NR3>,<NR3>,..."

## Examples

SENSe:NOISE:FREQUENCY:TABLE:NEW "1.2e9,1.5e9,1.452e9" sets 3 step values in a new Frequency Table to 1.200 GHz, 1.350 GHz, and 1.500 GHz.

## [SENSe]:NR:ACP:SWEep:TIME

Sets or queries the sweep time of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:SWEEp:TIME <value>
```

```
[SENSe]:NR:ACP:SWEEp:TIME?
```

## Related Commands

## Arguments

<value> : : <NRf> specifies the sweep time in ms

## Returns

<NRf>

## Examples

SENSe:NR:ACP:SWEEP:TIME? might return 1.0000000000E-3, indicating the sweep time value is 1.0000000000E-3 seconds.

## [SENSe]:NR:ACP:SWEEp:TIME:AUTO

Sets or queries the sweep time checkbox state of NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:SWEEp:TIME:AUTO {OFF | ON | 0 | 1 }
```

```
[SENSe]:NR:ACP:SWEEp:TIME:AUTO?
```

## Arguments

OFF or 0 disables sweep time.

ON or 1 enables sweep time.

## Returns

0 means sweep time is disabled.

1 means sweep time is enabled.

## Examples

SENSe:NR:ACP:SWEEP:TIME:AUTO OFF disables sweep time.

## [SENSe]:NR:ACP:FFT:WINDow

Sets or queries the FFT Window of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:FFT:WINDow {NONE | FLATTOP | BLACKMANHARRIS | HANNING | HAMMING | GAUSSIAN | BLACKMAN | KAISERBESSEL}
```

```
[SENSe]:NR:ACP:FFT:WINDow?
```

## Arguments

The following arguments set the FFT window to the given transform:

NONE

FLATTOP

BLACKMANHARRIS

HANNING

HAMMING

GAUSSIAN

BLACKMAN

KAISERBESSEL

## Returns

See Arguments

## Examples

`SENSe:NR:ACP:FFT:WINDOW?` might return `FLATTOP`, indicating that the FFT window is set to Flat Top.

## [SENSe]:NR:ACP:RBW

Sets or queries the RBW of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:RBW <value>
```

```
[SENSe]:NR:ACP:RBW?
```

## Arguments

<value>::<NRf> specifies the RBW

## Returns

<NRf>

## Examples

SENSe:NR:ACP:RBW? might return 30.000000000E+3, indicating the RBW value is 30.000000000E+3 Hz.

## [SENSe]:NR:ACP:RBW:AUTO

Sets or queries the RBW checkbox state of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

[SENSe]:NR:ACP:RBW:AUTO {OFF | ON | 0 | 1}

[SENSe]:NR:ACP:RBW:AUTO?

## Arguments

OFF or 0 disables RBW.

ON or 1 enables RBW.

## Returns

0 means RBW is disabled.

1 means RBW is enabled.

## Examples

SENSe:NR:ACP:RBW:AUTO OFF disables RBW.

## [SENSe]:NR:ACP:RBWFilter:TYPE

Sets or queries the RBW Filter type of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:RBWFilter:TYPE {FLAT | FFTBASED | GAUSSIAN}
```

```
[SENSe]:NR:ACP:RBWFilter:TYPE?
```

## Arguments

FLAT sets the RBW Filter to Flat

FFTBASED sets the RBW Filter to FFT Based

GAUSSIAN sets the RBW Filter to Gaussian

## Returns

See Arguments

## Examples

SENSe:NR:ACP:RBWFILTER:TYPE? might return FLAT, indicating that the RBW Filter is set to Flat.

## [SENSe]:NR:ACP:NO:OF:NR:OFFsets

Sets or queries the number of offsets of the NR ACP measurement.

## Conditions

Measurement view: NR Adjacent Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:ACP:NO:OF:NR:OFFsets <value>
```

```
[SENSe]:NR:ACP:NO:OF:NR:OFFsets?
```

## Arguments

<value>::<NR1> specifies the number of offsets.

## Returns

<NR1>

## Examples

SENSe:NR:ACP:NO:OF:NR:OFFSETS? might return 2, indicating the number of offset value is 2.

## [SENSe]:NR:AVEraging:AVEraging

Sets or queries the Averaging check box state common to all the 5GNR measurements.



## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:AVeraging:AVeraging { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Averaging.

ON or 1 enables Averaging.

## Returns

0 means Averaging is disabled.

1 means Averaging is enabled.

## Examples

```
SENSe:NR:AVERAGING:AVERAGING OFF disables Averaging.
```

```
SENSe:NR:AVERAGING:AVERAGING? might return 0, indicating that Averaging is Off and the Averaging check box is unchecked.
```

## [SENSe]:NR:AVeraging:NUMAVg

Sets or queries the Averaging count common to all the 5GNR measurements.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:AVeraging:NUMAVg <value>
```

```
[SENSe]:NR:AVeraging:NUMAVg?
```

## Arguments

<value>::<NR1> specifies the averaging count.

## Returns

<NR1>

## Examples

`SENSe:NR:AVERAGING:NUMAVG?` might return 5, indicating that the averaging count is set to 5.

## [SENSe]:NR:AVEraging:CHP:TYPE

It sets or queries CHP Averaging type for the NR CHP measurement.

### Conditions

Measurement views: All 5GNR measurement views

### Group

Sense commands

### Syntax

```
SENSe:NR:AVEraging:CHP:TYPE{ RMS | LOG | SCALAR | MAX | MIN }
```

```
SENSe:NR:AVEraging:CHP:TYPE? (Query)
```

### Arguments

RMS sets the CHP Averaging type to RMS.

LOG sets the CHP Averaging type to Log.

SCALAR sets the CHP Averaging type to Scalar.

MAX sets CHP Averaging type to Max.

MIN sets the CHP Averaging type to Min.

### Examples

`SENSe:NR:AVEraging:CHP:TYPE?` might return RMS, indicating that the Averaging type is set to RMS.

## [SENSe]:NR:AVEraging:ACP:TYPE

It sets or queries ACP Averaging type for the NR ACP measurement.

### Conditions

Measurement views: All 5GNR measurement views

### Group

Sense commands

### Syntax

```
SENSe:NR:AVEraging:ACP:TYPE{ RMS | LOG | SCALAR | MAX | MIN }
```

```
SENSe:NR:AVEraging:ACP:TYPE? (Query)
```

## Arguments

RMS sets the ACP Averaging type to RMS.

LOG sets the ACP Averaging type to Log.

SCALAR sets the ACP Averaging type to Scalar.

MAX sets the ACP Averaging type to Max.

MIN sets the ACP Averaging type to Min.

## Examples

`SENSe:NR:AVeraging:ACP:TYPe?` might return RMS, indicating that the Averaging type is set to RMS.

## [SENSe]:NR:AVeraging:SEM:TYPe

It sets or queries SEM Averaging type for the NR SEM measurement.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:AVeraging:SEM:TYPe{ RMS | LOG | SCALAR | MAX | MIN }`

`SENSe:NR:AVeraging:SEM:TYPe? (Query)`

## Arguments

RMS sets the SEM Averaging type to RMS.

LOG sets the SEM Averaging type to Log.

SCALAR sets the SEM Averaging type to Scalar.

MAX sets SEM Averaging type to Max.

MIN sets the SEM Averaging type to Min.

## Examples

`SENSe:NR:AVeraging:SEM:TYPe?` might return RMS, indicating that the SEM Averaging type is set to RMS.

## [SENSe]:NR:AVeraging:OBW:TYPe

It sets or queries OBW Averaging type for the NR OBW measurement.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:AVeraging:OBW:TYPe{ RMS | LOG | SCALAR | MAX | MIN }
```

```
SENSe:NR:AVeraging:OBW:TYPe? (Query)
```

## Arguments

RMS sets the OBW Averaging type to RMS.

LOG sets the OBW Averaging type to Log.

SCALAR sets the OBW Averaging type to Scalar.

MAX sets OBW Averaging type to Max.

MIN sets the OBW Averaging type to Min.

## Examples

SENSe:NR:AVeraging:OBW:TYPe? might return RMS, indicating that the OBW Averaging type is set to RMS.

## [SENSe]:NR:AVeraging:PVT:TYPe

It sets or queries PVT Averaging type for the NR PVT measurement.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:AVeraging:PVT:TYPe{ RMS | LOG | SCALAR | MAX | MIN }
```

```
SENSe:NR:AVeraging:PVT:TYPe? (Query)
```

## Arguments

RMS sets the PVT Averaging type to RMS.

LOG sets the PVT Averaging type to Log.

## Examples

SENSe:NR:AVeraging:PVT:TYPe? might return RMS, indicating that the PVT Averaging type is set to RMS.

## [SENSe]:NR:CELL:ID:AUTO

Sets or queries the state of the Cell ID check box for displays common to all the 5GNR measurements. On disabling, adds the Cell ID column to the Carrier Configuration grid for manual configuration.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSE]:NR:CELL:ID:AUTO { 0 | 1 | OFF | ON }
```

```
[SENSE]:NR:CELL:ID:AUTO?
```

## Arguments

OFF or 0 disables automatic mode for Cell ID and enables manual mode for configuration of Cell ID.

ON or 1 enables automatic mode for Cell ID.

## Returns

0 means automatic mode for Cell ID is disabled and manual mode is enabled.

1 means automatic mode for Cell ID is enabled.

## Examples

SENSE:NR:CELL:ID:AUTO OFF enables manual mode to configure Cell ID.

## [SENSE]:NR:CHP:SWEep:TIME

Sets or queries the sweep time of the NR CHP measurement.

## Conditions

Measurement view: NR Channel Power

## Group

Sense commands

## Syntax

```
[SENSE]:NR:CHP:SWEep:TIME <value>
```

```
[SENSE]:NR:CHP:SWEep:TIME?
```

## Arguments

<value>::<NRf> specifies the sweep time (ms).

## Returns

<NRf>

## Examples

`SENSe:NR:CHP:SWEEP:TIME?` might return `1.0000000000E-3`, indicating the sweep time value is `1.0000000000E-3` s.

## **[SENSe]:NR:CHP:SWEEP:TIME:AUTO**

Sets or queries the sweep time checkbox state of the NR CHP measurement.

### Conditions

Measurement view: NR Channel Power

### Group

Sense commands

### Syntax

`[SENSe]:NR:CHP:SWEEP:TIME:AUTO {OFF | ON | 0 | 1}`

`[SENSe]:NR:CHP:SWEEP:TIME:AUTO?`

### Arguments

OFF or 0 disables sweep time.

ON or 1 enables sweep time.

### Returns

0 means sweep time is disabled.

1 means sweep time is enabled.

## Examples

`SENSe:NR:CHP:SWEEP:TIME:AUTO OFF` disables sweep time.

## **[SENSe]:NR:CHP:FFT:WINDOW**

Sets or queries the FFT Window of the NR CHP measurement.

### Conditions

Measurement view: NR Channel Power

### Group

Sense commands

### Syntax

`[SENSe]:NR:CHP:FFT:WINDOW { NONE | FLATTOP | BLACKMANHARRIS | HANNING | HAMMING | GAUSSIAN | BLACKMAN | KAISERBESSEL }`

`[SENSe]:NR:CHP:FFT:WINDOW?`

## Arguments

The following arguments set the FFT window to the given transform:

NONE

FLATTOP

BLACKMANHARRIS

HANNING

HAMMING

GAUSSIAN

BLACKMAN

KAISERBESSEL

## Returns

See Arguments

## Examples

`SENSE:NR:CHP:FFT:WINDOW?` might return `FLATTOP`, indicating that the FFT Window is set to Flat Top.

## [SENSe]:NR:CHP:INTEgration:BANDwidth:TYPE

Sets or queries the integration bandwidth type of the NR CHP measurement.

## Conditions

Measurement view: NR Channel Power

## Group

Sense commands

## Syntax

```
[SENSe]:NR:CHP:INTEgration:BANDwidth:TYPE {SIGNALBANDWIDTH | CHANNELBANDWIDTH}
```

```
[SENSe]:NR:CHP:INTEgration:BANDwidth:TYPE?
```

## Arguments

SIGNALBANDWIDTH

CHANNELBANDWIDTH

## Returns

See Arguments

## Examples

`SENSE:NR:CHP:INTEGRATION:BANDWIDTH:TYPE?` might return `SIGNALBANDWIDTH`, indicating that the integration bandwidth is set to Signal Bandwidth.

## **[SENSe]:NR:CHP:RBW**

Sets or queries the RBW of the NR CHP measurement.

### **Conditions**

Measurement view: NR Channel Power

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:CHP:RBW <value>
```

```
[SENSe]:NR:CHP:RBW?
```

### **Arguments**

<value>::<NRf> specifies the RBW

### **Returns**

<NRf>

### **Examples**

SENSe:NR:CHP:RBW? might return 30.000000000E+3, indicating the RBW value is 30.000000000E+3 Hz.

## **[SENSe]:NR:CHP:RBW:AUTO**

Sets or queries the RBW checkbox state of the NR CHP measurement.

### **Conditions**

Measurement view: NR Channel Power

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:CHP:RBW:AUTO {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:CHP:RBW:AUTO?
```

### **Arguments**

OFF or 0 disables RBW.

ON or 1 enables RBW.

### **Returns**

0 means RBW is disabled.



1 means RBW is enabled.

## Examples

`SENSe:NR:CHP:RBW:AUTO OFF` disables RBW.

## [SENSe]:NR:CHP:RBWFilter:TYPE

Sets or queries the RBW filter type of the NR CHP measurement.

### Conditions

Measurement view: NR Channel Power

### Group

Sense commands

### Syntax

```
[SENSe]:NR:CHP:RBWFilter:TYPE {FLAT| FFTBASED | GAUSSIAN}
```

```
[SENSe]:NR:CHP:RBWFilter:TYPE?
```

### Arguments

FLAT

FFTBASED

GAUSSIAN

### Returns

See Arguments

### Examples

`SENSe:NR:CHP:RBWFILTER:TYPE?` might return `FLAT`, indicating that the RBW Filter is set to Flat.

## [SENSe]:NR:DOWNlink:CHANnel:CONFig:MODE

Sets or queries the downlink channel configuration mode of all of the 5GNR measurements.

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

```
[SENSe]:NR:DOWNlink:CHANnel:CONFig:MODE { TEST | USER }
```

```
[SENSe]:NR:DOWNlink:CHANnel:CONFig:MODE?
```

## Arguments

TEST sets the downlink channel configuration mode to Test Model.

USER sets the downlink channel configuration mode to User Defined.

## Returns

See Arguments

## Examples

SENSe:NR:DOWNLINK:CHANNEL:CONFIG:MODE? might return USER, indicating that the downlink channel configuration mode is set to User Defined.

## [SENSe]:NR:EVM:TRAcE:DRAwing

It sets or queries the trace drawing type of "5G NR EVM per symbol trace view".

## Conditions

Measurement views: EVM 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:EVM:TRAcE:DRAwing {VECTORS,POINTS}
```

```
SENSe:NR:EVM:TRAcE:DRAwing? (Query)
```

## Arguments

VECTORS sets the trace drawing type to Vector.

POINTS sets the trace drawing type to Points.

## Examples

SENSe:NR:EVM:TRAcE:DRAwing VECTORS sets the trace drawing type to Vector.

## [SENSe ]:NR:EVM:TRAcE:SELEction

It sets or queries the trace type of "5G NR EVM per symbol trace view".

## Conditions

Measurement views: EVM 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:EVM:TRAcE:SELection {RMSSYMBOLMEAN,,PEAKSYMBOLMAX}}
```

```
SENSe:NR:EVM:TRAcE:SELection? (Query)
```

## Arguments

RMSSYMBOLMEAN will set trace type to “RMS EVM per Symbol Mean Trace”.

PEAKSYMBOLMAX will set the trace type to “Peak EVM per Symbol Maximum Trace”.

## Examples

```
SENSe:NR:EVM:TRAcE:SELection RMSSYMBOLMEAN sets the trace display type to “RMS EVM per Symbol Mean Trace”.
```

## [SENSe]:NR:FREquency:AUTO

Sets or queries the state of the Frequency Offset check box for all displays common to the 5GNR measurements. On disabling, the command adds the Frequency Offset column to the Carrier Configuration grid.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:FREquency:AUTO {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:FREquency:AUTO?
```

## Arguments

OFF or 0 disables Frequency Offset.

ON or 1 enables Frequency Offset.

## Returns

0 means Frequency Offset is disabled.

1 means Frequency Offset is enabled.

## Examples

```
SENSe:NR:FREQUENCY:AUTO
```

## [SENSe]:NR:GNODEb:CATEgory

Sets or queries the gNodeB Category common to all of the 5GNR measurements.

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:GNODEb:CATegory { WABSCATEGORYA | WABSCATEGORYBOPTION1 |  
WABSCATEGORYBOPTION2 | LABS | MRBS }
```

```
[SENSe]:NR:GNODEb:CATegory?
```

## Arguments

WABSCATEGORYA, sets the gNodeB Category set to A Wide Area Base Station

WABSCATEGORYBOPTION1, sets the gNodeB Category set to B(Optional) Wide Area BS

WABSCATEGORYBOPTION2, sets the gNodeB Category set to B(Optional) Wide Area BS

LABS, sets gNodeB Category set to Local Area Base Station

MRBS, sets the gNodeB Category set to Medium Range Base Station

## Returns

See Arguments

## Examples

[SENSe]:NR:GNODEB:CATegory? might return MRBS, indicating that the gNodeB Category is set to Medium Range Base Station.

## [SENSe]:NR:LINK:DIREction

Sets or queries the Link direction type for all the 5G NR measurements.

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:LINK:DIREction {DOWN | UPLI}
```

```
[SENSe]:NR:LINK:DIREction?
```

## Arguments

DOWN sets the Link direction to Downlink UPLI sets the Link direction to Uplink

## Returns

See Arguments

## Examples

`SENSe:NR:LINK:DIRECTION DOWN` sets the Link direction to Downlink.

`SENSe:NR:LINK:DIRECTION?` might return UPLI, indicating that the Link direction is set to Uplink.

## [SENSe]:NR:MODulation:MEAs:LENgth

Sets or queries the measurement length of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

`[SENSe]:NR:MODulation:MEAs:LENgth <value>`

`[SENSe]:NR:MODulation:MEAs:LENgth?`

## Arguments

`<value> : : <NRf>` specifies the measurement time.

## Returns

`<NRf>`

## Examples

`[SENSe]:NR:MODULATION:MEAS:LENGTH?` might return 1.0000000000, indicating the measurement time value is 1.0000000000 ms.

## [SENSe]:NR:MODulation:MEAs:OFFset

Sets or queries the measurement offset of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

`[SENSe]:NR:MODulation:MEAs:OFFset <value>`

[SENSe]:NR:MODulation:MEAs:OFFset?

### Arguments

<alue>: : <NRf> specifies the measurement offset.

### Returns

<NRf>

### Examples

SENSe:NR:MODULATION:MEAS:OFFSET? might return 1.0000000000, indicating the measurement offset value is 1.0000000000 s.

## [SENSe]:NR:MODulation:MEAs:LENgth:UNIt

Sets or queries measurement unit of the NR Constellation measurement.

### Conditions

Measurement view: NR Constellation

### Group

Sense commands

### Syntax

[SENSe]:NR:MODulation:MEAs:LENgth:UNIt {SLOT | TIME}

[SENSe]:NR:MODulation:MEAs:LENgth:UNIt?

### Arguments

SLOT

TIME

### Returns

See Arguments

### Examples

SENSe:NR:MODULATION:MEAS:LENGTH:UNIT? might return Slot, indicating that the measurement unit is set to Slot.

## [SENSe]:NR:MODulation:FFT:WINDow:LENgth

Sets or queries the FFT Window Length of the NR Constellation measurement.

### Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:FFT:WINDow:LENgth <value>
```

```
[SENSe]:NR:MODulation:FFT:WINDow:LENgth?
```

## Arguments

<value> : : <NRf> specifies the measurement time (ms).

## Returns

<NRf>

## Examples

SENSe:NR:MODULATION:FFT:WINDOW:LENGTH? might return 1.0000000000, indicating the FFT Window Length value is 1.0000000000 s.

## [SENSe]:NR:MODulation:FFT:WINDow:LENgth:AUTO

Sets or queries the FFT Window Length checkbox state of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:FFT:WINDow:LENgth:AUTO {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:FFT:WINDow:LENgth:AUTO?
```

## Arguments

OFF or 0 disables FFT Window Length ON or 1 enables FFT Window Length

## Returns

0 means FFT Window Length is disabled

1 means FFT Window Length is enabled

## Examples

SENSe:NR:MODULATION:FFT:WINDOW:LENGTH:AUTO OFF disables FFT Window Length.

## **[SENSe]:NR:MODulation:FFT:WINDow:TYPE**

Sets or queries FFT Window Type of the NR Constellation measurement.

### **Conditions**

Measurement view: NR Constellation

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:MODulation:FFT:WINDow:TYPE {GPP3 | CUSTOM}
```

```
[SENSe]:NR:MODulation:FFT:WINDow:TYPE?
```

### **Arguments**

GPP3

CUSTOM

### **Returns**

See Arguments

### **Examples**

`SENSe:NR:MODULATION:FFT:WINDOW:TYPE?` might return GPP3, indicating that the FFT Window is set to 3GPP.

## **[SENSe]:NR:MODulation:SPECTrum:FLATness**

Sets or queries the spectral flatness type of the NR Constellation measurement.

### **Conditions**

Measurement view: NR Constellation

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:MODulation:SPECTrum:FLATness {NORMAL | EXTREME}
```

```
[SENSe]:NR:MODulation:SPECTrum:FLATness?
```

### **Arguments**

NORMAL

EXTREME



## Returns

See Arguments

## Examples

[SENSE]:NR:MODULATION:SPECTRUM:FLATNESS? might return NORMAL, indicating that the spectral flatness type is set to Normal.

## [SENSe]:NR:MODulation:DC:SUBcarrier:REMOval

Sets or queries the DC SubCarrier Removal checkbox state of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:DC:SUBcarrier:REMOval {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:DC:SUBcarrier:REMOval?
```

## Arguments

OFF or 0 disables DC SubCarrier Removal ON or 1 enables DC SubCarrier Removal

## Returns

0 means that DC SubCarrier Removal is disabled.

1 means that DC SubCarrier Removal is enabled.

## Examples

SENSe:NR:MODULATION:DC:SUBCARRIER:REMOVAL OFF disables DC SubCarrier Removal.

## [SENSe]:NR:MODulation:SPECTrum:INVErted

Sets or queries the Spectrum Inverted checkbox state of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:SPECTrum:INVErted {OFF | ON | 0 | 1}
```

[SENSe]:NR:MODulation:SPECTrum:INVErted?

### Arguments

OFF or 0 disables Spectrum Inverted ON or 1 enables Spectrum Inverted

### Returns

0 means that Spectrum Inverted is disabled.

1 means that Spectrum Inverted is enabled.

### Examples

SENSe:NR:MODULATION:SPECTRUM:INVERTED OFF disables Spectrum Inverted.

## [SENSe]:NR:MODulation:PHase:TRAcking

Sets or queries the Phase Tracking checkbox state of the NR Constellation measurement

### Conditions

Measurement view: NR Constellation

### Group

Sense commands

### Syntax

[SENSe]:NR:MODulation:PHase:TRAcking {OFF | ON | 0 | 1}

[SENSe]:NR:MODulation:PHase:TRAcking?

### Arguments

OFF or 0 disables Phase Tracking

ON or 1 enables Phase Tracking

### Returns

0 means that Phase Tracking is disabled.

1 means that Phase Tracking is enabled.

### Examples

SENSe:NR:MODULATION:PHASE:TRACKING OFF disables Phase Tracking.

## [SENSe]:NR:MODulation:TIMing:TRAcking

Sets or queries the Timing Tracking checkbox state of the NR Constellation measurement

### Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:TIMing:TRAcking {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:TIMing:TRAcking?
```

## Arguments

OFF or 0 disables Timing Tracking

ON or 1 enables Timing Tracking

## Returns

0 means that Timing Tracking is disabled.

1 means that Timing Tracking is enabled.

## Examples

```
SENSe:NR:MODULATION:TIMING:TRACKING ON enables Timing Tracking.
```

## [SENSe]:NR:MODulation:EVM:UNITs

Sets or queries EVM Units Type of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:EVM:UNITs {PERCENT | DB}
```

```
[SENSe]:NR:MODulation:EVM:UNITs?
```

## Arguments

PERCENT sets the EVM Units to Percent DB sets the EVM Units to dB

## Returns

See Arguments

## Examples

```
SENSe:NR:MODULATION:EVM:UNITs? might return PERCENT, indicating that the EVM Units Type is set to Percent.
```

## **[SENSe]:NR:MODulation:INClude:DMRs**

Sets or queries the Include DMRS checkbox state of NR Constellation measurement.

### **Conditions**

Measurement view: NR Constellation

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:MODulation:INClude:DMRs {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:INClude:DMRs?
```

### **Arguments**

OFF or 0 disables Include DMRS

ON or 1 enables Include DMRS

### **Returns**

0 means that Include DMRS is disabled.

1 means that Include DMRS is enabled.

### **Examples**

```
SENSe:NR:MODULATION:INCLUDE:DMRS OFF disables Include DMRS.
```

## **[SENSe]:NR:MODulation:ESTimation:CHAnnel**

Sets or queries Estimation Channel Type of the NR Constellation measurement.

### **Conditions**

Measurement view: NR Constellation

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:MODulation:ESTimation:CHAnnel {DDATA | DMRS}
```

### **Arguments**

DDATA sets the Estimation Channel to DMRS and Data.

DMRS set the Estimation Channel to DMRS.

## Returns

See Arguments

## Examples

`SENSe:NR:MODULATION:ESTIMATION:CHANNEL?` might return `DMRS`, indicating that the Estimation Channel type is set to `DMRS`.

## [SENSe]:NR:MODulation:ESTimation:FREquency:ERRor

Sets or queries the Frequency Error Type of the NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:ESTimation:FREquency:ERRor {NORMAL | DISABLED | WIDE}
```

```
[SENSe]:NR:MODulation:ESTimation:FREquency:ERRor?
```

## Arguments

`NORMAL`

`DISABLED`

`WIDE`

## Returns

See Arguments

## Examples

`SENSe:NR:MODULATION:ESTIMATION:FREQUENCY:ERROR?` might return `NORMAL`, indicating that the Frequency Error is set to `Normal`.

## [SENSe]:NR:MODulation:Estimation:COMmon:CLOck:SOUrce

Sets or queries the Common clock source checkbox state of NR CONSTELLATION measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:MODulation:Estimation:COMmon:CLOck:SOUrce { OFF | ON | 0 | 1 }  
[SENSe]:NR:MODulation:Estimation:COMmon:CLOck:SOUrce? (Query)
```

## Arguments

OFF or 0 disables Common Clock Source.

ON or 1 enables Common Clock Source.

## Returns

0 disables Common Clock Source.

1 enables Common Clock Source.

## Examples

```
SENSe:NR:MODULATION:Estimation:COMMON:CLOCK:SOURCE OFF, disables Common clock source.
```

## [SENSe]:NR:MODulation:Estimation:IQ:ORIGIN:OFFset

Sets or queries the IQ Origin Offset checkbox state of the NR CONSTELLATION measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands

## Syntax

```
SENSe:NR:MODulation:ESTimation:IQ:ORIGIN:OFFset? (Query)  
SENSe:NR:MODulation:ESTimation:IQ:ORIGIN:OFFset { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables IQ Origin Offset

ON or 1 enables IQ Origin Offset

## Returns

0 disables IQ Origin Offset.

1 enables IQ Origin Offset.

## Examples

```
SENSe:NR:MODulation:ESTimation: IQ:ORIGIN:OFFset OFF, disables IQ Origin Offset.
```

## [SENSe]:NR:MODulation:ESTimation:IQ:IMPairments

Sets or queries IQ impairments checkbox state of the NR Constellation measurement.

### Conditions

Measurement view: NR Constellation

### Group

Sense commands

### Syntax

```
[SENSe]:NR:MODulation:ESTimation:IQ:IMPairments {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:ESTimation:IQ:IMPairments?
```

### Arguments

OFF or 0 disables IQ impairments

ON or 1 enables IQ impairments

### Returns

0 means IQ impairments is disabled

1 means IQ impairments is enabled

### Examples

```
SENSe:NR:MODULATION:ESTIMATION:IQ:IMPAIRMENTS 0, disables IQ impairments.
```

## [SENSe]:NR:MODulation:ESTimation:SYMBOL:CLOCK:ERROR

Sets or queries Symbol Clock Error checkbox state of the NR Constellation measurement

### Conditions

Measurement view: NR Constellation

### Group

Sense commands

### Syntax

```
[SENSe]:NR:MODulation:ESTimation:SYMBOL:CLOCK:ERROR {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:MODulation:ESTimation:SYMBOL:CLOCK:ERROR?
```

### Arguments

OFF or 0 disables Symbol Clock Error ON or 1 enables Symbol Clock Error

## Returns

0 means Symbol Clock Error is disabled

1 means Symbol Clock Error is enabled

## Examples

`SENSe:NR:MODULATION:ESTIMATION:SYMBOL:CLOCK:ERROR 1`, enables Symbol Clock Error.

## [SENSe]:NR:MODulation:INClude:PTRs

Sets or queries the Include PTRs checkbox state of NR Constellation measurement.

## Conditions

Measurement view: NR Constellation

## Group

Sense commands

## Syntax

`[SENSe]:NR:MODulation:INClude:PTRs {OFF | ON | 0 | 1}`

`[SENSe]:NR:MODulation:INClude:PTRs?`

## Arguments

OFF or 0 disables Include PTRs.

ON or 1 enables Include PTRs.

## Returns

0 means that Include PTRs is disabled.

1 means that Include PTRs is enabled.

## Examples

`SENSe:NR:MODULATION:INCLUDE:PTRs OFF` disables Include PTRs.

## [SENSe]:NR:MODulation:TRACe:CC:INDEX

Sets or queries the Component Carrier Index of NR Constellation measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands



## Syntax

```
SENSe:NR:MODulation:TRACe:CC:INDEx<value>  
SENSe:NR:MODulation:TRACe:CC:INDEx? (Query)
```

## Argument

<value>::<Nrf> specifies the Measurement Time(ms).

## Returns

<Nrf>

## Examples

SENSe:NR:MODulation:TRACe:CC:INDEx? Might return 1, which indicates the Component Carrier Index value is 1.

## [SENSe]:NR:MODulation:TRACe:PTRS

Sets or queries Trace PTRS checkbox state of NR CONSTELLATION measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands

## Syntax

```
SENSe:NR:MODulation:TRACe:PTRS? (Query)  
SENSe:NR:MODulation:TRACe:PTRS{ OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Trace PTRS.

ON or 1 enables Trace PTRS.

## Returns

0 disables Trace PTRS.

1 enables Trace PTRS.

## Examples

SENSe:NR:MODulation:TRACe:PTRS OFF, disables Trace PTRS.

## [SENSe]:NR:MODulation:TRACe:DMRS

Sets or queries Trace DMRS checkbox state of NR CONSTELLATION measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands.

## Syntax

```
SENSe:NR:MODulation:TRACe:DMRS? (Query)
```

```
SENSe:NR:MODulation:TRACe:DMRS{ OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Trace DMRS

ON or 1 enables Trace DMRS.

## Returns

0 disables Trace DMRS .

1 enables Trace DMRS .

## Examples

```
NSe:NR:MODulation:TRACe:DMRS OFFSE, disables Trace DMRS.
```

## [SENSe]:NR:MODulation:TRACe:DATA

Sets or queries Trace Data checkbox state of NR CONSTELLATION measurement.

## Conditions

Measurement view: NR Constellation measurement view

## Group

Sense commands

## Syntax

```
SENSe:NR:MODulation:TRACe:DATA? (Query)
```

```
SENSe:NR:MODulation:TRACe:DATA{ OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Trace data

ON or 1 enables Trace data.

## Returns

0 disables Trace data .

1 enables Trace data .

## Examples

`SENSe:NR:MODulation:TRACe:DATA OFF`, disables Trace data.

## [SENSe]:NR:MODulation:FFT:WINDow:OFFset

Sets or queries the FFT Window OFFset of NR Constellation measurement.

### Conditions

Measurement view: NR Constellation measurement view

### Group

Sense commands

### Syntax

`SENSe:NR:MODulation:FFT:WINDow:OFFset <value>`

`SENSe:NR:MODulation:FFT:WINDow:OFFset? (Query)`

### Argument

`<value>::<NRf>` specifies the FFT window offset.

### Returns

`<NRf>`

## Examples

`SENSe:NR:MODulation:FFT:WINDow:OFFset?` Might return 50.0000000000, which indicates the FFT Window OFFset value is 50.0000000000 .

## [SENSe]:NR:OBW:FFTWIndow:TYPE

It sets or queries the FFT Window of the NR OBW measurement.

### Conditions

Measurement view: NR Occupied Bandwidth measurement view

### Group

Sense commands

### Syntax

`SENSe:NR:OBW:FFTWIndow:TYPE { NONE | FLATTOP | BLACKMANHARRIS | HANNING | HAMMING | GAUSSIAN | BLACKMAN | KAISERBESSEL }`

`SENSe:NR:OBW:FFTWIndow:TYPE? (Query)`

## Arguments

NONE sets the FFT Window to None.

FLATTOP sets the FFT Window to Flat Top.

BLACKMANHARRIS sets the FFT Window to Blackman-Harris.

HANNING sets the FFT Window to Hanning.

HAMMING sets the FFT Window to Hamming.

GAUSSIAN sets the FFT Window to Gaussian.

BLACKMAN sets the FFT Window to Blackman.

KAISERBESSEL sets the FFT Window to Kaiser Bessel.

## Examples

`SENSe:NR:OBW:FFTWindow:TYPE?` might return FLATTOP, indicating that the FFT Window is set to Flat Top.

## [SENSe]:NR:OBW:LIMIt

Sets or queries the Limit of NR OBW measurement.

## Conditions

Measurement view: NR Occupied Bandwidth measurement view

## Group

Sense commands

## Syntax

`SENSe:NR:OBW:LIMIt <value>`

`cSENSe:NR:OBW:LIMIt? (Query)`

## Argument

`<value>::<NRf>` specifies the limit.

## Returns

`<NRf>`

## Examples

`SENSe:NR:OBW:LIMIt?` Might return `30.000000000E+3`, which indicates the Limit value is `30.000000000E+3` Hz.

## [SENSe]:NR:OBW:LIMIt:AUTo

Sets or queries the RBW checkbox state of NR OBW measurement.

## Conditions

Measurement view: NR Occupied Bandwidth measurement view

## Group

Sense commands

## Syntax

```
SENSe:NR:OBW:LIMIt:AUTo? (Query)
```

```
SENSe:NR:OBW:LIMIt:AUTo { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables LIMIt.

ON or 1 enables LIMIt.

## Returns

0 disables LIMIt.

1 enables LIMIt.

## Examples

```
SENSe:NR:OBW:LIMIt:AUTo OFF disables LIMIt.
```

## [SENSe]:NR:OBW:MEAS:BW

It sets or queries Measurement Bandwidth value of the NR OBW measurement.

## Conditions

Measurement view: NR Occupied Bandwidth measurement view

## Group

Sense commands

## Syntax

```
SENSe:NR:OBW:MEAS:BW<value>
```

```
SENSe:NR:OBW:MEAS:BW? (Query)
```

## Argument

<value>::<NRf> specifies the meas BW.

## Returns

<NRf>

## Examples

`SENSe:NR:OBW:MEAS:BW?` Might return 20000000, which indicates the Measurement Bandwidth BW value is 200.00 Mhz.

## [SENSe]:NR:OBW:RBW

Sets or queries the RBW of NR OBW measurement.

### Conditions

Measurement view: NR Occupied Bandwidth measurement view

### Group

Sense commands

### Syntax

`SENSe:NR:OBW:RBW<value>`

`SENSe:NR:OBW:RBW? (Query)`

### Argument

`<value>::<NRf>` specifies the RBW.

### Returns

`<NRf>`

## Examples

`SENSe:NR:OBW:RBW?` Might return 30.0000000000E+3, which indicates the RBW value is 30.000 kHz.

## [SENSe]:NR:OBW:RBW:AUTO

Sets or queries the RBW checkbox state of NR OBW measurement.

### Conditions

Measurement view: NR Occupied Bandwidth measurement view

### Group

Sense commands

### Syntax

`SENSe:NR:OBW:RBW:AUTO? (Query)`

`SENSe:NR:OBW:RBW:AUTO { OFF | ON | 0 | 1 }`

### Arguments

OFF or 0 disables RBW.

ON or 1 enables RBW.

### Returns

0 disables RBW.

1 enables RBW.

### Examples

`SENSe:NR:OBW:RBW:AUTO OFF` disables RBW.

## [SENSe]:NR:OBW:RBWFilter:TYPE

It sets or queries the RBW Filter type of the NR OBW measurement.

### Conditions

Measurement view: NR Occupied Bandwidth measurement view

### Group

Sense commands

### Syntax

```
SENSe:NR:OBW:RBWFilter:TYPE{ FLAT| FFTBASED | GAUSSIAN}
```

```
SENSe:NR:OBW:RBWFilter:TYPE? (Query)
```

### Arguments

FLAT sets the RBW Filter to Flat.

FFTBASED sets the RBW Filter to FFT Based.

GAUSSIAN sets the RBW Filter to Gaussian.

### Examples

`SENSe:NR:OBW:RBWFilter:TYPE?` might return FLAT, indicating that the RBW Filter is set to Flat.

## [SENSe]:NR:OBW:SWEEptime

Sets or queries the Sweep time of NR OBW measurement.

### Conditions

Measurement view: NR Occupied Bandwidth measurement view

### Group

Sense commands

## Syntax

SENSe:NR:OBW:SWEEptime <value>

SENSe:NR:OBW:SWEEptime? (Query)

## Argument

<value>::<NRf> specifies the Sweep Time(ms).

## Returns

<NRf>

## Examples

SENSe:NR:OBW:SWEEptime? Might return 1.0000000000E-3, which indicates the sweep time value is 1.0000000000E-3 sec.

## [SENSe]:NR:OBW:SWEEptime:AUTO

Sets or queries the Sweep time checkbox state of NR OBW measurement.

## Conditions

Measurement view: NR Occupied Bandwidth measurement view

## Group

Sense commands

## Syntax

SENSe:NR:OBW:SWEEptime:AUTO? (Query)

SENSe:NR:OBW:SWEEptime:AUTO { OFF | ON | 0 | 1 }

## Arguments

OFF or 0 disables Sweep time.

ON or 1 enables Sweep time.

## Returns

0 disables Sweep time.

1 enables Sweep time.

## Examples

SENSe:NR:OBW:SWEEptime:AUTO OFF disables Sweep Time.

## [SENSe]:NR:PVT:OFFPower:AFTRExclusion

Sets or queries the OFF Power Exclusion After of NR PVT measurement.



## Conditions

Measurement view: NR Power vs Time view

## Group

Sense commands

## Syntax

```
SENSe:NR:PVT:OFFPower:AFTRExclusion <value>
```

```
SENSe:NR:PVT:OFFPower:AFTRExclusion? (Query)
```

## Argument

<value>::<NRf> specifies the OFF Power Exclusion After.

## Examples

SENSe:NR:PVT:OFFPower:AFTRExclusion? Might return 1.000000000E-3, which indicates the OFF Power Exclusion After value is 1.000000000E-3 sec.

## [SENSe]:NR:PVT:OFFPower:BFRExclusion

Sets or queries the OFF Power Exclusion Before of NR PVT measurement.

## Conditions

Measurement view: NR Power vs Timeview

## Group

Sense commands

## Syntax

```
SENSe:NR:PVT:OFFPower:BFRExclusion <value>
```

```
SENSe:NR:PVT:OFFPower:BFRExclusion? (Query)
```

## Argument

<value>::<NRf> specifies the OFF Power Exclusion Before.

## Examples

SENSe:NR:PVT:OFFPower:BFRExclusion? Might return 1.000000000E-3, which indicates the OFF Power Exclusion Before value is 1.000000000E-3 sec.

## [SENSe]:NR:PVT:APPLy:LIMIt

Sets or queries the Apply limit checkbox state of NR PVT measurement.

## Conditions

Measurement view: NR Power vs Timeview

## Group

Sense commands

## Syntax

```
SENSe:NR:PVT:APPLY:LIMIt? (Query)
```

```
SENSe:NR:PVT:APPLY:LIMIt{ OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Apply Limit.

ON or 1 enables Apply Limit.

## Returns

0 disables Apply Limit

1 enables Apply Limit.

## Examples

```
SENSe:NR:PVT:APPLY:LIMIt OFF disables Apply Limit.
```

## [SENSe]:NR:RESOource:BLOCK:AUTO

Sets or queries the state of the Resource block check box for displays common to all the 5GNR measurements. On disabling, adds the Resource block column to the Carrier Configuration grid for manual configuration.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:RESOource:BLOCK:AUTO { 0 | 1 | OFF | ON }
```

```
[SENSe]:NR:RESOource:BLOCK:AUTO?
```

## Arguments

OFF or 0 disables automatic mode for Resource block and enables manual mode for configuration of the Resource block.

ON or 1 enables automatic mode for the Resource block.

## Returns

0 means automatic mode for the Resource block is disabled and manual mode in enabled.

1 means automatic mode for the Resource block is enabled.

## Examples

`SENSe:NR:RESOURCE:BLOCK:AUTO OFF` enables manual mode to configure the Resource block.

## [SENSe]:NR:REFErence:GRID:AUTO

Sets or queries the state of the Reference grid check box for displays common to all the 5GNR measurements. On disabling, adds the Reference grid column to the Carrier Configuration grid for manual configuration.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:REFErence:GRID:AUTO { 0 | 1 | OFF | ON }
```

```
[SENSe]:NR:REFErence:GRID:AUTO?
```

## Arguments

OFF or 0 disables automatic mode for Reference grid and enables manual mode for configuration of the Reference grid.

ON or 1 enables automatic mode for the Reference grid.

## Returns

0 means automatic mode for the Reference grid is disabled and manual mode in enabled.

1 means automatic mode for the Reference grid is enabled.

## Examples

`SENSe:NR:REFERENCE:GRID:AUTO OFF` enables manual mode to configure the Reference grid.

## [SENSe]:NR:SEM:SWEep:TIME

Sets or queries the Sweep time of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SEM:SWEep:TIME <value>
```

[SENSe]:NR:SEM:SWEep:TIME?

## Arguments

<value>::<NRf> specifies the sweep time.

## Returns

<NRf>

## Examples

SENSe:NR:SEM:SWEEP:TIME might return 1.0000000000E-3, indicating the sweep time value is 1 s.

## [SENSe]:NR:SEM:SWEep:TIME:AUTO

Sets or queries the Sweep time checkbox state of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

[SENSe]:NR:SEM:SWEep:TIME:AUTO {OFF | ON | 0 | 1}

[SENSe]:NR:SEM:SWEep:TIME:AUTO?

## Arguments

OFF or 0 disables sweep time

ON or 1 enables sweep time

## Returns

0 means sweep time is disabled

1 means sweep time is enabled

## Examples

SENSe:NR:SEM:SWEEP:TIME:AUTO OFF disables sweep time.

## [SENSe]:NR:SEM:FFT:WINdow

Sets or queries FFT window of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SEM:FFT:WINDow {NONE | FLATTOP | BLACKMANHARRIS | HANNING | HAMMING | GAUSSIAN | BLACKMAN | KAISERBESSEL}
```

```
[SENSe]:NR:SEM:FFT:WINDow?
```

## Arguments

The following arguments set the FFT window to the given transform:

NONE

FLATTOP

BLACKMANHARRIS

HANNING

HAMMING

GAUSSIAN

BLACKMAN

KAISERBESSEL

## Returns

See Arguments

## Examples

`SENSe:NR:SEM:FFT:WINDOW?` might return `FLATTOP`, indicating that the FFT window is set to Flat Top.

## [SENSe]:NR:SEM:UPLInk:MASK:TYPE

Sets or queries Uplink Mask type of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SEM:UPLInk:MASK:TYPE {GENERAL | NS_35 | CUSTOM | NS_3 | NS_4 | NS_6 | NS_21}
```

```
[SENSe]:NR:SEM:UPLInk:MASK:TYPE?
```

## Arguments

GENERAL

NS\_35

CUSTOM

NS\_3

NS\_4

NS\_6

NS\_21

## Returns

See Arguments

## Examples

`SENSe:NR:SEM:UPLINK:MASK:TYPE?` might return GENERAL, indicating that the Uplink Mask is set to General.

## [SENSe]:NR:SEM:DOWNlink:MASK:TYPE

Sets or queries Downlink Mask type of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

`[SENSe]:NR:SEM:DOWNlink:MASK:TYPE {STANDARD | CUSTOM}`

`[SENSe]:NR:SEM:DOWNlink:MASK:TYPE?`

## Arguments

STANDARD

CUSTOM

## Returns

See Arguments

## Examples

`SENSe:NR:SEM:DOWNLINK:MASK:TYPE?` might return CUSTOM, indicating that the Downlink Mask is set to Custom.

## [SENSe]:NR:SEM:DELTaf:MAXimum

Sets or queries the maximum frequency deviation of the SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SEM:DELTAf:MAXImum <value>
```

```
[SENSe]:NR:SEM:DELTAf:MAXImum?
```

## Arguments

<value>::<NRf> specifies the maximum frequency deviation.

## Returns

<NRf>

## Examples

SENSe:NR:SEM:DELTAf:MAXIMUM? might return 15.000000000E+6, indicating the maximum frequency deviation is set to 15 MHz.

## [SENSe]:NR:SEM:RBW

Sets or queries the RBW of the NR SEM measurement.

## Conditions

Measurement view: NR Spectral Emission Mask

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SEM:RBW <value>
```

```
[SENSe]:NR:SEM:RBW?
```

## Arguments

<value>::<NRf> specifies the RBW.

## Returns

<NRf>

## Examples

SENSe:NR:SEM:RBW? might return 30.000000000E+3, indicating the RBW value is 30 kHz.

## **[SENSe]:NR:SEM:RBW:AUTO**

Sets or queries the RBW checkbox state of the NR SEM measurement.

### **Conditions**

Measurement view: NR Spectral Emission Mask

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:SEM:RBW:AUTO {OFF | ON | 0 | 1}
```

```
[SENSe]:NR:SEM:RBW:AUTO?
```

### **Arguments**

OFF or 0 disables RBW

ON or 1 enables RBW

### **Returns**

0 means RBW is disabled

1 means RBW is enabled

### **Examples**

```
SENSe:NR:SEM:RBW:AUTO OFF disables RBW.
```

## **[SENSe]:NR:SEM:RBWFilter:TYPE**

Sets or queries RBW Filter type of the NR SEM measurement.

### **Conditions**

Measurement view: NR Spectral Emission Mask

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:SEM:RBWFilter:TYPE {FLAT | FFTBASED | GAUSSIAN}
```

```
[SENSe]:NR:SEM:RBWFilter:TYPE?
```

### **Arguments**

FLAT

FFTBASED

GAUSSIAN



## Returns

See Arguments

## Examples

`SENSe:NR:SEM:RBWFILTER:TYPE?` might return `FLAT`, indicating that the RBW Filter is set to Flat.

## [SENSe]:NR:SUBblock<x>:BAND

Sets or queries the band parameter common to all the 5GNR measurements. `x=1`, which is the range of the Subblock

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

`[SENSe]:NR:SUBblock<x>:BAND <value>`

`[SENSe]:NR:SUBblock<x>:BAND?`

## Arguments

`<value>::<NR1>` which specifies the band parameter

## Returns

`<NR1>`

## Examples

`SENSe:NR:SUBblock1:BAND 86` will set the band parameter of subblock 1 to 86.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BANDwidth

Sets or queries the measurement carrier bandwidth of all the 5GNR measurements.

`<x> = 1`, which is the subblock range.

`<y>` is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

`[SENSe]:NR:SUBblock<x>:CARrier<y>:BANDwidth <value>`

[SENSe]:NR:SUBblock<x>:CARrier<y>:BANDwidth?

### Arguments

<value>: : <NRf> specifies the carrier bandwidth

### Returns

<NRf>

### Examples

SENSe:NR:SUBblock1:CARrier6:BANDwidth 100000000 sets the bandwidth for subblock 1 carrier 2 to 100 MHz.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:CYCLic:PREFix

Sets or queries the Bandwidth Part cyclic prefix type common to all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:CYCLic:PREFix{NORMAL}

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:CYCLic:PREFix?

### Arguments

NORMAL

### Returns

See Arguments

### Examples

SENSe:NR:SUBblock1:CARrier3:BWP1:CYCLic:PREFix? might return NORMAL, indicating that the cyclic prefix for subblock 1, carrier 3, BWP 1 is set to Normal.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:GRID:START

Sets or queries the measurement Bandwidth Part grid start of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSE]:NR:SUBblock<x>:CARrier<y>:BWP<z>:GRID:START <value>
```

```
[SENSE]:NR:SUBblock<x>:CARrier<y>:BWP<z>:GRID:START?
```

## Arguments

<value>::<NRf> specifies the grid start

## Returns

<NRf>

## Examples

SENSE:NR:SUBblock1:CARrier3:BWP1:GRID:START 1 sets the grid start for subblock 1, carrier 3, BWP 1 to 1.

## [SENSE]:NR:SUBblock<x>:CARrier<y>:BWP<z>:SUBCarrier:SPACing

Sets or queries the measurement carrier Bandwidth Part subcarrier spacing of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSE]:NR:SUBblock<x>:CARrier<y>:BWP<z>:SUBCarrier:SPACing{FIFTEEN | THIRTY | SIXTY}
```

```
[SENSE]:NR:SUBblock<x>:CARrier<y>:BWP<z>:SUBCarrier:SPACing?
```

## Arguments

FIFTEEN sets the subcarrier spacing to 15 kHz THIRTY sets the subcarrier spacing to 30 kHz SIXTY sets the subcarrier spacing to 60 kHz

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier3:BWP1:SUBCarrier:SPACing?` might return FIFTEEN, indicating that the subcarrier spacing for subblock 1, carrier 3, BWP 1 is 15 kHz.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK:OFFSet

Sets or queries the measurement carrier Bandwidth Part Resource Block offset of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK: OFFSet <value>

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK:OFFSet?

## Arguments

<value>::<NRf> specifies the Resource Block offset

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier3:BWP1:RESOurce:BLOCK:OFFSet 2` sets the Resource Block offset for subblock 1, carrier 3, BWP 1 to 2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK: NUMBRs

Sets or queries the measurement Bandwidth Part Resource Block number of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK: NUMBRS <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:RESOurce:BLOCK:NUMBRS?
```

## Arguments

<value>::<NRf> specifies the Resource Block number

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:RESOurce:BLOCK:NUMBRS 2 sets the Resource Block number for subblock 1, carrier 2, BWP 1 to 2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH: DMRS:ANTports

Sets or queries the PDSCH antenna ports of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH: DMRS:ANTports <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH: DMRS:ANTports?
```

## Arguments

<value>::<NRf> specifies the PDSCH antenna ports

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:ANTports 0` sets the PDSCH antenna ports for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:ADDIpos**

Sets or queries the PDSCH mapping additional positions of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:ADDIpos {ZERO | ONE | TWO | THREE}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:ADDIpos?
```

## Arguments

ZERO

ONE

TWO

THREE

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:MAPping:ADDIpos ZERO` sets the PDSCH mapping additional positions for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:CONFtype**

Sets or queries the PDSCH mapping configuration type of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:CONFtype {TYPE1 | TYPE2}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:CONFtype?
```

## Arguments

TYPE1

TYPE2

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping:CONFtype TYPE1 sets the PDSCH mapping configuration type for subblock 1, carrier 2, BWP 1 to Type 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:DU Ration

Sets or queries the PDSCH mapping duration of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:DURation {SINGLEsymbol | DOUBLEsymbol}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:DURation?
```

## Arguments

SINGLEsymbol

DOUBLEsymbol

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:MAPping:DURation SINGLEsymbol sets the PDSCH mapping duration for subblock 1, carrier 2, BWP 1 to Single Symbol.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:TYPE

Sets or queries the PDSCH mapping type of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH: DMRS: MAPping:TYPE {TYPEA | TYPEB}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH: DMRS:MAPping:TYPE?
```

## Arguments

TYPEA

TYPEB

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:MAPping:TYPE TYPEB sets the PDSCH mapping type for subblock 1, carrier 2, BWP 1 to Type B.



## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:TYPEApos**

Sets or queries the PDSCH mapping Type A position of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

### **Group**

Sense commands

### **Syntax**

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:TYPEApos <value>

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:MAPping:TYPEApos?

### **Arguments**

<value> : : <NRf> specifies the PDSCH mapping Type A position

### **Returns**

<NRf>

### **Examples**

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:MAPping:TYPEApos 0 sets the PDSCH mapping Type A position for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:NUMcdm**

Sets or queries the PDSCH number of CDM groups of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

### **Group**

Sense commands

### **Syntax**

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:NUMcdm <value>

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:NUMcdm?

### Arguments

<value>::<NRf> specifies the PDSCH number of CDM groups

### Returns

<NRf>

### Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:NUMcdm 1 sets the PDSCH number of CDM groups for subblock 1, carrier 2, BWP 1 to 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:POWmode

Sets or queries the PDSCH power mode of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:POWmode {CDM | USER}

[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:POWmode?

### Arguments

CDM sets the PDSCH Power mode to CDM.

USER sets the PDSCH Power mode to User Defined.

### Returns

See Arguments

### Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:POWmode? might return CDM, indicating that the PDSCH Power mode for subblock 1, carrier 2, BWP 1 is set to CDM.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:ID**

Sets or queries the PDSCH scrambling ID of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:ID <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:ID?
```

### **Arguments**

<value>::<NRf> specifies the PDSCH scrambling ID

### **Returns**

<NRf>

### **Examples**

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:SCRAmbling:ID 0` sets the PDSCH scrambling ID for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:INIT**

Sets or queries the PDSCH scrambling initialization of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

### **Group**

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:INIT <value>  
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:INIT?
```

## Arguments

<value>::<Nrf> specifies the PDSCH scrambling initialization

## Returns

<Nrf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:SCRAmbling:INIT 0 sets the PDSCH scrambling initialization for subblock 1, carrier 2, BWP 1 to 0.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:MODE

Sets or queries the PDSCH scrambling mode of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:MODE {CELLid |  
USERdefined}  
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:SCRAmbling:MODE?
```

## Arguments

CELLid

USERdefined

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:SCRAMbling:MODE CELLid` sets the PDSCH scrambling mode for subblock 1, carrier 2, BWP 1 to Cell ID.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:UDPower**

Sets or queries the Bandwidth Part PDSCH power value of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:UDPower <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:DMRS:UDPower?
```

## Arguments

<value>::<NRf> specifies the PDSCH power value

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:DMRS:UDPower?` might return 0, indicating that the PDSCH Power for subblock 1, carrier 2, BWP 1 is set to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:MODtype**

Sets or queries the PDSCH modulation type of the specified component carrier for all 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:MODtype{QPSK|QAM16|QAM64|QAM256|QAM1024}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:MODtype?
```

## Arguments

QPSK, QAM16, QAM64, QAM256, QAM1024

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:MODtype QPSK` sets the PDSCH modulation type for subblock 1, carrier 2, BWP 1 to QPSK.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ENABLEd

Sets or queries the PDSCH PTRS checkbox state for the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ENABLEd? (Query)
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ENABLEd { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables PTRS.

ON or 1 enables PTRS.

## Returns

0 disables PTRS.

1 enables PTRS.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:ENABled OFF`, disables PTRS Checkbox for carrier 2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:EPRE**

Sets or queries the PDSCH Energy per resource Element Ratio of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:EPRE {ZERO|ONE}`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:EPRE? (Query)`

## Arguments

ZERO sets the Energy per resource Element Ratio to 0.

ONE sets the Power mode to Energy per resource Element Ratio to 1.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:EPRE ZERO`, sets the Energy per resource Element Ratio to 0 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:FR Eqdensity**

Sets or queries the PDSCH Frequency density of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:FREqdensity { 2|4 }  
SENSe:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:FREqdensity? (Query)
```

## Arguments

2 sets the Frequency density to 2.

4 sets the Frequency density to 4.

## Examples

`SENSe:NR:SUBBlock1:CARrier2:BWP1:USER:PDSCH:PTRS:MAPping:FREqdensity 2`, sets the Frequency density to 00 for carrier2.

## [SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:REOffset

Sets or queries the PDSCH Resource Element offset of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:REOffset { 0|1|2|3 }  
SENSe:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:REOffset? (Query)
```

## Arguments

0 sets the Resource Element offset to 00.

1 sets the Resource Element offset to 01.

2 sets the Resource Element offset to 10.

3 sets the Resource Element offset to 11.



## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:MAPping:REOffset 0`, sets the Resource Element offset to 00 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:TIMedensity**

Sets or queries the PDSCH Time density of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:TIMedensity{ 1|2|4 }`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:MAPping:TIMedensity? (Query)`

## Arguments

1 sets the Time density to 1.

2 sets the Time density to 2.

4 sets the Time density to 4.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:MAPping:TIMedensity 2`, sets the Time density to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:POWmode**

Sets or queries the PDSCH Power mode of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:POWmode{STANdard|  
USERdefined }
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:POWmode? (Query)
```

## Arguments

STANdard sets the Power mode to Standard.

USERdefined sets the Power mode to User Defined.

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:POWmode STANdard, sets the Power mode to Standard for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:TRANSform:PREcode:GRPnumber

Sets or queries the PDSCH Time density of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS: TRANSform:PREcode:GRPnumber
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:TRANSform:PREcode:GRPnumber?  
(Query)
```

## Arguments

2 sets the Group number to 2.

4 sets the Group number to 4.

8 sets the Group number to 8.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:PREcode:GRPnumber 2`, sets the Group number to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:TRANSform:PREcode:SAMples**

Sets or queries the PDSCH Sample number of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS: TRANSform:PREcode:SAMples`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:TRANSform:PREcode: SAMples?`  
(Query)

## Arguments

2 sets the Group number to 2.

4 sets the Group number to 4.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:PREcode:SAMples 2`,sets the Sample number to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ANTports**

Sets or queries the PDSCH Antenna ports for selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ANTports<value>
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:ANTports? (Query)
```

## Argument

<value> : <NRf> specifies the PDSCH Antenna ports.

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:PTRS:ANTports 1000, sets the Antenna ports to 1000 for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:UDPower

Sets or queries the PDSCH User Defined power for selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:UDPower<value>
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:UDPower? (Query)
```

## Argument

<value> : <NRf> specifies the PDSCH User Defined power.

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:UDPower 0`, sets the User Defined power to 0 dB for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALL**

Sets or queries the status of the PDSCH Slot Allocation check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALL{OFF | ON | 0 | 1}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALL?
```

## Arguments

OFF or 0 disables Slot Allocation

ON or 1 enables Slot Allocation

## Returns

OFF or 0 means Slot Allocation is disabled ON or 1 means Slot Allocation is enabled

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:SLOT:ALL OFF` disables slot allocation for subblock 1, carrier 2, BWP 1.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALLOcation**

Sets or queries the PDSCH Slot Allocation value of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALLOcation <string>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SLOT:ALLOcation?
```

## Arguments

<string> specifies PDSCH slot allocation

## Returns

<string>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SLOT: ALLOcation 1:14 sets the slot allocation for subblock 1, carrier 2, BWP 1:14.

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:SLOT: ALLOcation 1,2:7,9 sets the slot allocation for subblock 1, carrier 2, BWP 1, 2:7, 9.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL:ALL**

Sets or queries the status of the PDSCH Symbol Allocation check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL: ALL{OFF|ON|0|1}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL:ALL?
```

## Arguments

OFF or 0 disables Symbol Allocation ON or 1 enables Symbol Allocation

## Returns

OFF or 0 means Symbol Allocation is disabled ON or 1 means Symbol Allocation is enabled

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:SYMBOL:ALL OFF` disables symbol allocation for subblock 1, carrier 2, BWP 1. This allows you to enter supported symbols manually.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL:ALLOcatio n**

Sets or queries the PDSCH Symbol Allocation value of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL:ALLOcation <string>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:SYMBOL:ALLOcation?
```

## Arguments

<string> specifies PDSCH symbol allocation

## Returns

<string>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:SYMBOL:ALLOcation 1:14` sets the symbol allocation for subblock 1, carrier 2, BWP 1:14.

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PDSCH:SYMBOL:ALLOcation 1,2:7,9` sets the symbol allocation for subblock 1, carrier 2, BWP 1, 2:7, 9.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:ANTports**

Sets or queries the PUSCH antenna ports of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:ANTports <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:ANTports?
```

## Arguments

<value>::<NRf> specifies the PUSCH antenna ports

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:ANTports 0` sets the PUSCH antenna ports for subblock 1, carrier 2, BWP 1 to 0.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:AD DIpos

Sets or queries the PUSCH mapping additional positions of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:ADDIpos {ZERO | ONE | TWO | THREE}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:ADDIpos?
```

## Arguments

ZERO

ONE



TWO

THREE

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping:ADDIpos ZERO` sets the PUSCH mapping additional positions for subblock 1, carrier 2, BWP 1 to 0.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:CONFtype

Sets or queries the PUSCH mapping configuration type of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:CONFtype {TYPE1 | TYPE2}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:CONFtype?
```

## Arguments

TYPE1

TYPE2

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping: CONFtype TYPE1` sets the PUSCH mapping configuration type for subblock 1, carrier 2, BWP 1 to Type 1.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:DU Ration**

Sets or queries the PUSCH mapping duration of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:DURation  
{SINGLEsymbol | DOUBLEsymbol}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:DURation?
```

### **Arguments**

SINGLEsymbol

DOUBLEsymbol

### **Returns**

See Arguments

### **Examples**

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping: DURation SINGLEsymbol sets`  
the PUSCH mapping duration for subblock 1, carrier 2, BWP 1 to Single Symbol.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TY PE**

Sets or queries the PUSCH mapping type of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

### **Conditions**

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TYPE{TYPEA|TYPEB}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TYPE?
```

## Arguments

TYPEA

TYPEB

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping:TYPE TYPEB` sets the PUSCH mapping type for subblock 1, carrier 2, BWP 1 to Type B.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TYPEApos

Sets or queries the PUSCH mapping Type A position of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TYPEApos <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:MAPping:TYPEApos?
```

## Arguments

<value>::<NRf> specifies the PUSCH mapping Type A position

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:MAPping: TYPEApos 0` sets the PUSCH mapping Type A position for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:NUMcdm**

Sets or queries the PUSCH number of CDM groups of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

`[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:NUMcdm <value>`

`[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:NUMcdm?`

## Arguments

<value>::<NRf> specifies the PUSCH number of CDM groups

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:NUMcdm 1` sets the PUSCH number of CDM groups for subblock 1, carrier 2, BWP 1 to 1.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:POWmode**

Sets or queries the Bandwidth Part PUSCH power mode of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:POWmode {CDM | USER}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:POWmode?
```

## Arguments

CDM sets the PUSCH Power mode to CDM.

USER sets the PUSCH Power mode to User Defined.

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:POWmode?` might return CDM, indicating that the PUSCH Power mode for subblock 1, carrier 2, BWP 1 is set to CDM.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAMbling:ID

Sets or queries the PUSCH scrambling ID of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAMbling:ID <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAMbling:ID?
```

## Arguments

<value>::<NRf> specifies the PUSCH scrambling ID

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:SCRAmbling:ID 0` sets the PUSCH scrambling ID for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:INIT**

Sets or queries the PUSCH scrambling initialization of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:INIT <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:INIT?
```

## Arguments

<value>::<NRf> specifies the PUSCH scrambling initialization

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:SCRAmbling:INIT 0` sets the PUSCH scrambling initialization for subblock 1, carrier 2, BWP 1 to 0.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:MODE**

Sets or queries the PUSCH scrambling mode of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:MODE {CELLid | USERdefined}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:SCRAmbling:MODE?
```

## Arguments

CELLid

USERdefined

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:SCRAmbling:MODE CELLid sets the PUSCH scrambling mode for subblock 1, carrier 2, BWP 1 to Cell ID.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:GRPhop:ENABLEd

Sets or queries the status of the PUSCH group hopping check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:GRPhop:ENABLEd {OFF | 0 ON | 1}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:GRPhop:ENABLEd?
```

## Arguments

OFF or 0 disables Transform group hopping ON or 1 enables Transform group hopping

## Returns

OFF or 0 means Transform group hopping is disabled ON or 1 means Transform group hopping is enabled

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:TRANSform:GRPhop:EN OFF` disables transform group hopping for subblock 1, carrier 2, BWP 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ENABled

Sets or queries the status of the PUSCH Transform Precoding check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5G NR measurement view

## Group

Sense commands

## Syntax

`[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ENABled {OFF | 0 | ON | 1}`

`SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ENABled?`

## Arguments

OFF or 0 disables Transform Precoding ON or 1 enables Transform Precoding

## Returns

OFF or 0 means Transform Precoding is disabled ON or 1 means Transform Precoding is enabled

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:TRANSform:PREcode:ENAB OFF` disables transform precoding for subblock 1, carrier 2, BWP 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ID

Sets or queries the PUSCH Transform Precoding ID of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8



<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ID
<value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:ID?
```

## Arguments

<value>::<NRf> specifies the Transform Precoding ID

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:TRANSform:PREcode:I 0 sets the PUSCH transform precode ID for subblock 1, carrier 2, BWP 1 to 0.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:IDMode

Sets or queries the PUSCH Transform Precoding ID mode of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:IDMode
{CELLid | USERdefined}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:PREcode:IDMode?
```

## Arguments

CELLid

USERdefined

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:TRANSform:PREcode:IDMo CELLid sets the PUSCH transform precoder ID mode for subblock 1, carrier 2, BWP 1 to Cell ID.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:TRANSform:SEQhop:ENABLEd

Sets or queries the status of the PUSCH sequence hopping check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:TRANSform:SEQhop:ENABLEd  
{OFF | 0 ON | 1}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: DMRS:TRANSform:SEQhop:ENABLEd?
```

## Arguments

OFF or 0 disables Transform sequence hopping ON or 1 enables Transform sequence hopping

## Returns

OFF or 0 means Transform sequence hopping is disabled ON or 1 means Transform sequence hopping is enabled

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:TRANSform:SEQhop:ENABl OFF disables transform sequence hopping for subblock 1, carrier 2, BWP 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:UDPower

Sets or queries the Bandwidth Part PUSCH power value of the specified Component Carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:UDPower <value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:DMRS:UDPower?
```

## Arguments

<value>::<NRf> specifies the PUSCH power

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:DMRS:UDPower? might return 0, indicating that the PUSCH Power for subblock 1, carrier 2, BWP 1 is set to 0.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:MODtype

Sets or queries the PUSCH modulation type of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:MODtype{PIBY2BPSK | QPSK | QAM16 | QAM64 | QAM256 | QAM1024}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:MODtype?
```

## Arguments

Arguments are the following modulation types:

PIBY2BPSK

QPSK

QAM16

QAM64

QAM256

QAM1024

## Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:MODtype QPSK` sets the PUSCH modulation type for subblock 1, carrier 2, BWP 1 to QPSK.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ANTports

Sets or queries the PUSCH Antenna ports for selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ANTports<value>`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ANTports? (Query)`

## Argument

<value>::<NRf> specifies the PUSCH Antenna ports.

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:ANTport 0`, sets the Antenna ports to 0 for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ENABLEd

Sets or queries the PUSCH PTRS checkbox state for the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ENABLEd? (Query)`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:ENABLEd { OFF | ON | 0 | 1 }`

## Arguments

OFF or 0 disables PTRS.

ON or 1 enables PTRS.

## Returns

0 disables PTRS.

1 enables PTRS.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:ENABLEd OFF`, disables PTRS Checkbox for carrier 2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:FR Eqdensity

Sets or queries the PUSCH Frequency density of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:FREqdensity { 2|4 }  
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:FREqdensity? (Query)
```

## Arguments

2 sets the Frequency density to 2.

4 sets the Frequency density to 4.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:MAPping:FREqdensity 2`, sets the Frequency density to 00 for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:REOffset

Sets or queries the PUSCH Resource Element offset of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:REOffset { 0|1|2|3 }  
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: PTRS:MAPping:REOffset? (Query)
```

## Arguments

0 sets the Resource Element offset to 00.

1 sets the Resource Element offset to 01.

2 sets the Resource Element offset to 10.

3 sets the Resource Element offset to 11.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:MAPping:REOffset 0`, sets the Resource Element offset to 00 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:TIMedensity**

Sets or queries the PUSCH Time density of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:TIMedensity{ 1|2|4 }`

`SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:MAPping:TIMedensity? (Query)`

## Arguments

1 sets the Time density to 1.

2 sets the Time density to 2.

4 sets the Time density to 4.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:MAPping:TIMedensity 2`, sets the Time density to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:POWmode**

Sets or queries the PUSCH Power mode of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PDSCH:PTRS:POWmode {STAN|USER}
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:POWmode? (Query)
```

## Arguments

STANdard sets the Power mode to Standard.

USERdefined sets the Power mode to User Defined.

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:POWmode STANdard, sets the Power mode to Standard for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:GRPnumber

Sets or queries the PUSCH transform pre coding Group Number of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5GNR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:GRPnumber  
{2|4|8}
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:GRPnumber?  
(Query)
```

## Arguments

2 sets the Group Number to 2.

4 sets the Group Number to 4.

8 sets the Group Number to 8.



## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:TRANSform:PREcode:GRPnumber 2`, sets the Group number to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:SAMples**

Sets or queries the PUSCH transform pre coding Samples of the selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:SAMples
{2|4|}
```

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:TRANSform:PREcode:SAMples?
(Query)
```

## Arguments

2 sets the Samples to 2.

4 sets the Samples to 4.

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:TRANSform:PREcode:SAMples 2`, sets the Samples to 2 for carrier2.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:UDPower**

Sets or queries the PUSCH User Defined power for selected Component Carrier.

The parameter <x>=1, representing range of subblock.

The parameter <y>=1 to 8, representing range of Component Carrier.

The parameter <z>=1, representing range of Bandwidth Part.

## Conditions

Measurement views: All 5G NR measurement views

## Group

Sense commands

## Syntax

```
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:UDPower<value>  
SENSe:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:PTRS:UDPower? (Query)
```

## Argument

<value> : : <NRf> specifies the PUSCH User Defined power.

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:PTRS:UDPower 0, sets the User Defined power to 0 dB for carrier2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SLOT:ALL

Sets or queries the status of the PUSCH Slot Allocation check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SLOT:ALL{OFF | ON | 0 | 1}  
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SLOT: ALL?
```

## Arguments

OFF or 0 disables Slot Allocation

ON or 1 enables Symbol Allocation

## Returns

OFF or 0 means Slot Allocation is disabled ON or 1 means Symbol Allocation is enabled

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SLOT:ALL OFF` disables slot allocation for subblock 1, carrier 2, BWP 1. This allows you to enter supported slots manually.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SLOT:ALLOcation**

Sets or queries the PUSCH Slot Allocation value of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: SLOT:ALLOcation <string>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH: SLOT:ALLOcation?
```

## Arguments

<string> specifies PUSCH symbol allocation

## Returns

<string>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SLOT:ALLOcation 1:14` sets the slot allocation for subblock 1, carrier 2, BWP 1:14.

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SLOT:ALLOcation 1,2:7,9` sets the slot allocation for subblock 1, carrier 2, BWP 1, 2:7, 9.

## **[SENSe]:NR:SUBblock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SYMBOL:ALL**

Sets or queries the status of the PUSCH Symbol Allocation check box of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SYMBOL: ALL {OFF | ON | 0 | 1}  
[SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SYMBOL:ALL ?
```

## Arguments

OFF or 0 disables Symbol Allocation ON or 1 enables Symbol Allocation

## Returns

OFF or 0 means Symbol Allocation is disabled ON or 1 means Symbol Allocation is enabled

## Examples

SENSe:NR:SUBBlock1:CARrier2:BWP1:USER:PUSCH:SYMBOL:ALL OFF disables symbol allocation for subblock 1, carrier 2, BWP 1. This allows you to enter supported symbols manually.

## [SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PUSCH:SYMBOL:ALLOcation n

Sets or queries the PUSCH Symbol Allocation value of the specified component carrier.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

<z> = 1, which is the Bandwidth Part range

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PUSCH: SYMBOL: ALLOcation <string>  
[SENSe]:NR:SUBBlock<x>:CARrier<y>:BWP<z>:USER:PUSCH: SYMBOL:ALLOcation?
```

## Arguments

<string> specifies PUSCH symbol allocation

## Returns

<string>

## Examples

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SYMBOL:ALLOcation 1:14` sets the symbol allocation for subblock 1, carrier 2, BWP 1:14.

`SENSe:NR:SUBblock1:CARrier2:BWP1:USER:PUSCH:SYMBOL:ALLOcation 1,2:7,9` sets the symbol allocation for subblock 1, carrier 2, BWP 1, 2:7, 9.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:CELL:ID

Sets or queries the measurement carrier Cell ID of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

`[SENSe]:NR:SUBblock<x>:CARrier<y>:CELL:ID <value>`

`[SENSe]:NR:SUBblock<x>:CARrier<y>:CELL:ID?`

## Arguments

<value>::<NRf> specifies the Cell ID

## Returns

<NRf>

## Examples

`SENSe:NR:SUBblock1:CARrier6:CELL:ID 2` sets the Cell ID for subblock 1 carrier 6 to 2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:DUPLex:SCHEme

Sets or queries the Duplex Scheme common to all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:DUPLex:SCHEme {DLFDD | DLTDD}  
[SENSe]:NR:SUBblock<x>:CARrier<y>:DUPLex:SCHEme?
```

## Arguments

DLFDD sets the duplex scheme to Downlink FDD DLTDD sets the duplex scheme to Downlink TDD

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:DUPLex:SCHEme? might return DLFDD, indicating that the duplex scheme is set to Downlink FDD for subblock 1 carrier 2.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:FREquency:OFFset

Sets or queries the measurement carrier frequency offset of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:FREquency:OFFset <value>  
[SENSe]:NR:SUBblock<x>:CARrier<y>:FREquency:OFFset?
```

## Arguments

<value>::<NRf> specifies the center frequency

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:FREquency:OFFset 1 sets the frequency offset of subblock 1 carrier 2 to 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:SPACing

Sets or queries the measurement carrier Reference Grid spacing of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:SPACing { FIFTEEN | THIRTY | SIXTY }
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:SPACing?
```

## Arguments

FIFTEEN sets the Reference grid spacing to 15 kHz THIRTY sets the Reference grid spacing to 30 kHz SIXTY sets the Reference grid spacing to 60 kHz

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier2:REFERence:GRID:SPACing? might return FIFTEEN, indicating that the Reference Grid spacing for subblock 1 carrier 2 is set to 15 kHz.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:START

Sets or queries the measurement carrier Reference Grid start of all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:START<value>
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:REFERence:GRID:START?
```

## Arguments

<value>: : <NRf> specifies the Reference Grid start

## Returns

<NRf>

## Examples

SENSe:NR:SUBblock1:CARrier2:REFerence:GRID:START 1 sets the Reference Grid start for subblock 1 carrier 2 to 1.

## [SENSe]:NR:SUBblock<x>:CARrier<y>:TEST:MODEl

Sets or queries the Test Model common to all the 5GNR measurements.

<x> = 1, which is the subblock range.

<y> is the Component Carrier. Range: 1–8

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:TEST:MODEl {TM1DOT1 | TM1DOT2 | TM2 | TM2A | TM3DOT1 | TM3DOT1A | TM3DOT2 | TM3DOT3}
```

```
[SENSe]:NR:SUBblock<x>:CARrier<y>:TEST:MODEl?
```

## Arguments

TM1DOT1 sets the Test Model to TM1.1

TM1DOT2 sets the Test Model to TM1.2

TM2 sets the Test Model to TM2

TM2A sets the Test Model to TM2A

TM3DOT1 sets the Test Model to TM3.1

TM3DOT1A sets the Test Model to TM3.1A

TM3DOT2 sets the sets the Test Model to TM3.2

TM3DOT3 sets the Test Model to TM3.3

## Returns

See Arguments

## Examples

SENSe:NR:SUBblock1:CARrier3:TEST:MODEl? might return TM1DOT1, indicating that the Test Model is set to TM1.1 for subblock 1 carrier 3.



## [SENSe]:NR:SUBblock<x>:CHANnel:RASTer

Sets or queries the channel raster common to all the 5GNR measurements. x=1, which is the range of the Subblock

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

```
[SENSe]:NR:SUBblock<x>:CHANnel:RASTer { FIFTEEN | SIXTY | HUNDRED }
```

### Arguments

FIFTEEN sets the Channel Raster to 15 kHz SIXTY sets the Channel Raster to 60 kHz HUNDRED sets the Channel Raster to 100 kHz

### Returns

See Arguments

### Examples

SENSe:NR:SUBblock1:CHANnel:RASTer FIFTEEN will set the channel raster of subblock 1 to 15 kHz.

SENSe:NR:SUBblock1:CHANnel:RASTer? might return SIXTY, indicating that the channel raster of subblock 1 to 60 kHz.

## [SENSe]:NR:SUBblock<x>:FREQuency:RANGe

Sets or queries the frequency range common to all the 5GNR measurements. x=1, which is the range of the Subblock

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

```
[SENSe]:NR:SUBblock<x>:FREQuency:RANGe { FR1 | FR2 }
```

```
[SENSe]:NR:SUBblock<x>:FREQuency:RANGe?
```

### Arguments

FR1 sets the frequency to Range1

FR2 sets the frequency to Range2

### Returns

See Arguments

## Examples

`SENSe:NR:SUBblock1:FREQuency:RANGe?` might return `FR1`, indicating that the Frequency Range is set to `Range1`.

## **[SENSe]:NR:SUBblock<x>:NUMCcs**

Sets or queries the number of component carriers common to all the 5GNR measurements.

`x=1`, which is the range of the Subblock

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

`[SENSe]:NR:SUBblock<x>:NUMCcs <value>`

`[SENSe]:NR:SUBblock<x>:NUMCcs?`

### Arguments

`<value>::<NR1>` specifies the number of carriers

### Returns

`<NR1>`

## Examples

`SENSe:NR:SUBblock1:NUMCcs 8` will set the number of component carriers of subblock 1 to 8.

## **SENSe:NR:SUBblock<x>:SELEcted:CC**

Sets or queries the Component Carriers selected in the Carrier Configuration tab. `x=1`, which is the range of the Subblock

### Conditions

Measurement view: Any 5GNR measurement view

### Group

Sense commands

### Syntax

`SENSe:NR:SUBblock<x>:SELEcted:CC`

### Arguments

`<value>::<NR1>`

## Returns

<NR1>

## Examples

`SENSe:NR:SUBBlock1:SELEcted:CC?` might return 5, indicating the selected component carrier of subblock 1 is 5.

Applies the selected parameters changes to all component carriers. common to all the 5GNR measurements.

`x=1`, which is the range of the Subblock.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:SUBBlock<x>: SELEcted:CC:APPLYtoall EXECUTE
```

## Examples

`SENSe:NR:SUBBlock1: SELEcted:CC:APPLYtoall EXECUTE`, apply selected changes to all Component carriers.

## [SENSe]:NR:TRANsmit:ARCHitec

Sets or queries the Transmitter Architecture common to all the 5GNR measurements.

## Conditions

Measurement view: Any 5GNR measurement view

## Group

Sense commands

## Syntax

```
[SENSe]:NR:TRANsmit:ARCHitec { LOCC | LOSB }
```

```
[SENSe]:NR:TRANsmit:ARCHitec?
```

## Arguments

LOCC sets the Transmitter Architecture set to LO Per Component Carrier LOSB sets the Transmitter Architecture set to LO Per Subblock

## Returns

See Arguments

## Examples

`SENSe:NR:TRANSMIT:ARCHITEC?` might return `LOSB`, indicating that the Transmitter Architecture is set to LO Per Subblock.

## [SENSe]:OBWidth:AVERage

Sets or queries whether to enable averaging in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:AVERage { OFF | ON | 0 | 1 }
```

```
[SENSe]:OBWidth:AVERage?
```

### Arguments

OFF or 0 disables averaging.

ON or 1 enables averaging.

## Examples

`SENSe:OBWIDTH:AVERAGE ON` enables averaging.

## [SENSe]:OBWidth:AVERage:COUNT

Sets or queries the number of measurements for averaging in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:AVERage:COUNT <number>
```

```
[SENSe]:OBWidth:AVERage:COUNT?
```

### Arguments

`<number>::<NR1>` specifies the average count. Range: 2 to 10000.

## Examples

`SENSe:OBWIDTH:AVERAGE:COUNT 64` sets the average count to 64.

## [SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement

Sets or queries the measurement bandwidth to determine the total power in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement <value>
```

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:MEASurement?
```

### Arguments

`<value>::<Nrf>` specifies the measurement bandwidth.

Range: 100 Hz to 109 MHz.

## Examples

`SENSe:OBWIDTH:BANDWIDTH:MEASUREMENT 10MHz` sets the measurement bandwidth to 10 MHz.

## [SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets `[SENSe]:OBWidth{BANDwidth|BWIDth}[:RESolution]:AUTO OFF`.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution] <value>
```

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]?
```

### Related Commands

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO
```

## Arguments

<value>::<Nrf> specifies the RBW. Range: 100 Hz to 5 MHz.

## Examples

SENSe:OBWIDTH:BANDWIDTH:RESOLUTION 200kHz sets the RBW to 200 kHz.

## [SENSe]:OBWidth:[BANDwidth|BWIDth][:RESolution]:ACTual? (Query Only)

Queries the actual resolution bandwidth (RBW) in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

## Arguments

None

## Returns

<Nrf> The actual RBW in Hz.

## Examples

SENSe:OBWIDTH:BANDWIDTH:RESOLUTION:ACTUAL? might return 299.624E+3, indicating that the actual RBW is 299.624 kHz.

## [SENSe]:OBWidth:[BANDwidth|BWIDth][:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

## Arguments

OFF or 0 specifies that the RBW is set manually using the `[SENSe]:OBWidth:{BANDwidth|BWIDth}[:RESolution]` command.

ON or 1 specifies that the RBW is set automatically.

## Examples

`SENSe:OBWIDTH:BANDWIDTH:AUTO ON` sets the RBW automatically.

## [SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets `[SENSe]:OBWidth{BANDwidth|BWIDth}:VIDeo:STATE` OFF.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo <value>
```

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo?
```

## Arguments

`<value>::<NRf>` specifies the VBW.

Range: Current RBW/104 (1 Hz minimum) to Current RBW.

## Examples

`SENSe:OBWIDTH:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz.

## [SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATE

Determines whether to enable the video bandwidth (VBW) in the Occupied Bandwidth measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATE { OFF | ON | 0 | 1 }
```

```
[SENSe]:OBWidth:{BANDwidth|BWIDth}:VIDeo:STATE?
```

## Arguments

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

## Examples

`SENSe:OBWIDTH:BANDWIDTH:VIDEO:STATE ON` enables the VBW.

## [SENSe]:OBWidth:BOBW:XDBM:MEASurement:DIREction

Sets or queries the measurement direction in the Bluetooth window.

## Conditions

Measurement views:Bluetooth 20 dB Bandwidth

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:BOBW:XDBM:MEASurement:DIREction {IN|OUT}
```

```
[SENSe]:OBWidth:BOBW:XDBM:MEASurement:DIREction?
```

## Arguments

IN specifies the inward direction

OUT specifies the outward direction.

## Examples

`SENSe:OBWIDTH:BOBW:XDBM:MEASUREMENT:DIRECTION?` might return IN indicating the measurement direction is set to IN.

## [SENSe]:OBWidth:BOBW:XDBM:RANGe

Sets or queries the measurement range in the Bluetooth window.

## Conditions

Measurement views:Bluetooth 20 dB Bandwidth

The x dBm option should be selected. If the x dB option is selected, the Range is always set to BOTH.

## Group

Sense commands

## Syntax

```
[SENSe]:OBWidth:BOBW:XDBM:RANGe {BOTH|LOW|HIGH}
```



---

[SENSe]:OBWidth:BOBW:XDBM:RANGe?

### Arguments

BOTH specifies both the high and low ranges.

LOW specifies the lower range.

HIGH specifies the higher range.

### Examples

SENSe:OBWIDTH:BOBW:XDBM:RANGE LOW Sets the measurement range to the lower range.

## [SENSe]:OBWidth:CLEar:RESults (No Query Form)

Restarts the average trace. This command is valid when [SENSe]:OBWidth: AVERage is set to ON.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

[SENSe]:OBWidth:CLEar:RESults

### Arguments

None

### Examples

SENSe:OBWIDTH:CLEAR:RESULTS restarts the average trace.

## [SENSe]:OBWidth:FREQuency:CENTer

Sets or queries the center frequency in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

[SENSe]:OBWidth:FREQuency:CENTer <value>

[SENSe]:OBWidth:FREQuency:CENTer?

## Arguments

`<value>::<NRf>` specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:OBWIDTH:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

## [SENSe]:OBWidth:FREQuency:STEP

Sets or queries the frequency step size in the Occupied Bandwidth measurement. Programming a specified step size sets `[SENSe]:OBWidth:FREQuency:STEP AUTO OFF`.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Sense commands

## Syntax

`[SENSe]:OBWidth:FREQuency:STEP <value>`

`[SENSe]:OBWidth:FREQuency:STEP?`

## Related Commands

`[SENSe]:OBWidth:FREQuency:STEP:AUTO`

## Arguments

`<value>::<NRf>` specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:OBWIDTH:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

## [SENSe]:OBWidth:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the Occupied Bandwidth measurement.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:FREQuency:STEP:AUTO{OFF|ON|0|1}
```

```
[SENSe]:OBWidth:FREQuency:STEP:AUTO?
```

### Arguments

OFF or 0 specifies that the frequency step size is set manually using the [SENSe]:OBWidth:FREQuency:STEP command.

ON or 1 specifies that the frequency step size is set automatically.

### Examples

SENSe:OBWIDTH:FREQUENCY:STEP:AUTO ON specifies that the frequency step size is set automatically.

## [SENSe]:OBWidth:PERCent

Sets or queries the occupied bandwidth percent power (power ratio of the occupied bandwidth to the measurement bandwidth).

### Conditions

Measurement views: Occupied Bandwidth

### Group

Sense commands

### Syntax

```
[SENSe]:OBWidth:PERCent <value>
```

```
[SENSe]:OBWidth:PERCent?
```

### Arguments

<value>::<NRf> specifies the occupied bandwidth percent power.

Range: 50 to 99.9%.

### Examples

SENSe:OBWIDTH:PERCENT 98 sets the occupied bandwidth percent power to 98%.

## **[SENSe]:OBWidth:XDBLevel**

Sets or queries the x dB level (how far down from the peak level the bandwidth is measured) in the x dB bandwidth measurement.

### **Conditions**

Measurement views: Occupied Bandwidth

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:OBWidth:XDBLevel <value>
```

```
[SENSe]:OBWidth:XDBLevel?
```

### **Arguments**

<value>::<NRf> specifies the x dB level. Range: -80 to -1 dB.

### **Examples**

SENSe:OBWIDTH:XDBLEVEL -10 sets the x dB level to -10 dB.

## **[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]**

Sets or queries the channel bandwidth in the OFDM view.

### **Conditions**

Measurement views: OFDM

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth] <value>
```

```
[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]?
```

### **Arguments**

<value>::<NRf> specifies the channel bandwidth.

### **Examples**

SENSe:OFDM:CHANNEL[:BANDWIDTH|BWIDTH] 20E+6 sets the channel bandwidth to 20 MHz.

## **[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]:SRATe? (Query Only)**

Returns the FFT sample rate for the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]:SRATe?
```

## Arguments

None

## Returns

<NRf> The sample rate in Hz.

## Examples

SENSe:OFDM:CHANNEL[:BANDWIDTH|BWIDTH]:SRATE? might return 20E+6 indicating the sample rate is 20 MHz.

## [SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]:STATe

Determines whether the channel bandwidth is on or off in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]:STATe{OFF|ON|0|1}
```

```
[SENSe]:OFDM:CHANnel[:BANDwidth|BWIDth]:STATe?
```

## Arguments

OFF or 0 specifies that the channel bandwidth is off.

ON or 1 specifies that the channel bandwidth is on.

## Examples

SENSe:OFDM:CHANNEL[:BANDWIDTH|BWIDTH]:STATE ON sets the channel bandwidth setting to on.

## [SENSe]:OFDM:CHANnel:ESTimation

Sets or queries the channel estimation in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:CHANnel:ESTimation{PREamble|PDATa}
```

```
[SENSe]:OFDM:CHANnel:ESTimation?
```

## Arguments

PREamble specifies the channel estimation is Preamble.

PDATa specifies the channel estimation is Preamble + Data.

## Examples

SENSe:OFDM:CHANNEL:ESTIMATION PDAT sets the channel estimation to Preamble + Data.

## [SENSe]:OFDM:CONStE:DETermination

Sets or queries the constellation determination in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:CONStE:DETermination{AUTO|MANual}
```

```
[SENSe]:OFDM:CONStE:DETermination?
```

## Arguments

AUTO specifies the instrument automatically detects the constellation determination.

MANual allows you to select a manual constellation type.

## Examples

SENSe:OFDM:CONStE:DETERMINATION MAN sets the constellation determination so you can select a manual constellation type using the SENSe:OFDM:CONStE:DETermination:MANual command.

## [SENSe]:OFDM:CONStE:DETermination:MANual

Sets or queries a manual constellation type. The Constellation determination must be set to Manual using the

```
[SENSe]:OFDM:CONStE:DETermination command.
```

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:CONStE:DETermination:MANual {BPSK|QAM16|QAM64|QPSK}
```

```
[SENSe]:OFDM:CONStE:DETermination:MANual?
```

## Related Commands

```
[SENSe]:OFDM:CONStE:DETermination
```

## Arguments

BPSK specifies the BPSK constellation type.

QAM16 specifies the QAM16 constellation type.

QAM64 specifies the QAM16 constellation type.

QPSK specifies the QPSK constellation type.

## Examples

```
SENSe:OFDM:CONStE:DETERMINATION:MANUAL QPSK selects the QPSK constellation type.
```

## [SENSe]:OFDM:FFT:LENGth? (Query Only)

Returns the FFT length for the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:FFT:LENGth?
```

## Arguments

None

## Returns

<NR1> The FFT length.

## Examples

`SENSe:OFDM:FFT:LENGTH?` might return the FFT length as 64.

## [SENSe]:OFDM:GUARd:INTerval

Sets or queries the guard interval in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:GUARd:INTerval {FOURth|EIGHth|SIXTeenth|THIRtysecond|USER}
```

### Arguments

`FOURth` specifies the guard interval is set to 1/4.

`EIGHth` specifies the guard interval is set to 1/8.

`SIXTeenth` specifies the guard interval is set to 1/16.

`THIRtysecond` specifies the guard interval is set to 1/32.

`USER` specifies a user-defined guard interval.

## Examples

`SENSe:OFDM:GUARd:INTERVAL THIR` sets the guard interval to 1/32.

## [SENSe]:OFDM:GUARd:INTerval:USER

Sets or queries a user-defined guard interval. The Guard Interval must be set to USER.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:GUARd:INTerval:USER <value>
```

```
[SENSe]:OFDM:GUARd:INTerval:USER?
```

### Related Commands

```
[SENSe]:OFDM:GUARd:INTerval
```



## Arguments

<value>::<Nrf> specifies the guard interval.

## Examples

SENSe:OFDM:GUARD:INTERVAL:USER 3.12 sets the guard interval to 3.12%.

## [SENSe]:OFDM:PILOt:TRACking:AMPLitude:STATe

Determines whether to select the Amplitude pilot tracking in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:PILOt:TRACking:AMPLitude:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:OFDM:PILOt:TRACking:AMPLitude:STATe?
```

## Arguments

OFF or 0 specifies that the Amplitude pilot tracking is off.

ON or 1 specifies that the Amplitude pilot tracking is on.

## Examples

SENSe:OFDM:PILOt:TRACking:AMPLitude:STATe ON turns the Amplitude pilot tracking on.

## [SENSe]:OFDM:PILOt:TRACking:PHASe:STATe

Determines whether to select the Phase pilot tracking in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:PILOt:TRACking:PHASe:STATe { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 specifies that the Phase pilot tracking is off.

ON or 1 specifies that the Phase pilot tracking is on.

## Examples

`SENSe:OFDM:PILOT:TRACKING:PHASE:STATE ON` turns the Phase pilot tracking on.

## **[SENSe]:OFDM:PILOt:TRACking:TIMing:STATe**

Determines whether to select the Timing pilot tracking in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:PILOt:TRACking:TIMing:STATe {OFF|ON|0|1}
```

### Arguments

OFF or 0 specifies that the Timing pilot tracking is off.

ON or 1 specifies that the Timing pilot tracking is on.

## Examples

`SENSe:OFDM:PILOT:TRACKING:TIMING:STATE ON` turns the Timing pilot tracking on.

## **[SENSe]:OFDM:RADix (No Query Form)**

Sets the radix for the OFDM measurements. This command is effective in the symbol table.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:RADix { BINary | HEX }
```

### Arguments

#### Returns

`BINary` specifies the binary radix.

`HEX` specifies the hexadecimal radix.

## Examples

`SENSe:OFDM:RADIX HEX` sets the radix to hexadecimal for the OFDM symbol table.

## [SENSe]:OFDM:SCARrier:SPACing

Sets or queries the subcarrier spacing in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SCARrier:SPACing <value>
```

```
[SENSe]:OFDM:SCARrier:SPACing?
```

### Arguments

`<value>::<NRf>` specifies the subcarrier spacing.

## Examples

`SENSe:OFDM:SCARRIER:SPACING 312.5E+3` sets the subcarrier spacing to 312.5 MHz.

## [SENSe]:OFDM:SCARrier:SPACing:STATe

Determines whether the subcarrier spacing is on or off in the OFDM view

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SCARrier:SPACing:STATe{OFF|ON|0|1}
```

```
[SENSe]:OFDM:SCARrier:SPACing:STATe?
```

### Arguments

OFF or 0 specifies that the subcarrier spacing is off.

ON or 1 specifies that the subcarrier spacing is on.

## Examples

`SENSe:OFDM:SCARRIER:SPACING:STATE ON` sets the subcarrier spacing to on.

## [SENSe]:OFDM:SCARriers

Sets or queries the Subcarriers to display in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SCARriers { ALL | PILOt | DATA | RANGE | SINGLE }  
[SENSe]:OFDM:SCARriers?
```

### Arguments

`ALL` specifies that all subcarriers will be displayed.

`PILOt` specifies that only the pilot subcarriers will be displayed.

`DATA` specifies that only the data subcarriers will be displayed.

`RANGE` specifies that the subcarriers within a specified range will be displayed.

`SINGLE` specifies that only a single subcarrier will be displayed.

## Examples

`SENSe:OFDM:SCARRIERS ALL` sets the OFDM view to display all subcarriers.

## [SENSe]:OFDM:SCARriers:RANGE:START

Sets or queries the subcarrier start range in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SCARriers:RANGE:START <value>  
[SENSe]:OFDM:SCARriers:RANGE:START?
```

## Related Commands

[SENSe]:OFDM:SCARriers:RANGe:STOP

## Arguments

<NR1>

## Examples

SENSe:OFDM:SCARRIERS:RANGE:START -20 sets the subcarrier start range to -20.

## [SENSe]:OFDM:SCARriers:RANGe:STOP

Sets or queries the subcarrier stop range in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

[SENSe]:OFDM:SCARriers:RANGe:STOP <value>

[SENSe]:OFDM:SCARriers:RANGe:STOP?

## Related Commands

[SENSe]:OFDM:SCARriers:RANGe:START

## Arguments

<NR1>

## Examples

SENSe:OFDM:SCARRIERS:RANGE:STOP 20 sets the subcarrier stop range to 20.

## [SENSe]:OFDM:SCARriers:SINGle:INDex

Sets or queries the index of a single subcarrier in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:SCARriers:SINGle:INDeX <Value>  
[SENSe]:OFDM:SCARriers:SINGle:INDeX?
```

## Arguments

<NR1>

## Examples

SENSe:OFDM:SCARRIERS:SINGLE:INDEX -10 sets the single subcarrier index to -10.

## [SENSe]:OFDM:STANdard

Sets or queries the OFDM standard.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:STANdard { S802_11 | S802_16 }  
[SENSe]:OFDM:STANdard?
```

## Arguments

S802\_11 specifies the 802.11a/g/j OFDM standard.

S802\_16 specifies the 802.16 (2004) standard.

## Examples

SENSe:OFDM:STANDARD S802\_16 sets the OFDM standard to 802.16.

## [SENSe]:OFDM:SWAP:IQ

Determines whether or not to exchange I and Q data before demodulating.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:SWAP:IQ{OFF|ON|0|1}
```

---

```
[SENSe]:OFDM:SWAP:IQ?
```

### Arguments

OFF or 0 uses I and Q data as they are.

ON or 1 exchanges I and Q data.

### Examples

```
SENSe:OFDM:SWAP:IQ ON exchanges I and Q data before demodulating.
```

## [SENSe]:OFDM:SYMBOL:ANALYSIS:OFFSET

Sets or queries the OFDM symbol analysis offset.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SYMBOL:ANALYSIS:OFFSET <value>
```

```
[SENSe]:OFDM:SYMBOL:ANALYSIS:OFFSET?
```

### Arguments

<value> specifies the symbol analysis offset as a percentage.

Range 0 to -100%.

### Examples

```
SENSe:OFDM:SYMBOL:ANALYSIS:OFFSET -50 sets the symbol analysis offset to -50%.
```

## [SENSe]:OFDM:SYMBOLS

Sets or queries the symbols to display in the OFDM view.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:SYMBOLS {ALL | RANGE | SINGLE}
```

```
[SENSe]:OFDM:SYMBOLS?
```

## Arguments

ALL specifies that all symbols will be displayed.

RANGE specifies that a range of symbols will be displayed.

SINGLE specifies that single symbol will be displayed.

## Examples

SENSE:OFDM:SYMBOLS ALL specifies that all symbols will be displayed in the OFDM views.

## [SENSe]:OFDM:SYMBOLs:MAX

Sets or queries the maximum number of symbols to analyze in the OFDM views.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:SYMBOLs:MAX <value>
```

```
[SENSe]:OFDM:SYMBOLs:MAX?
```

## Arguments

<NR1>

## Examples

SENSE:OFDM:SYMBOLS:MAX 50 specifies a maximum of 50 symbols to analyze in the OFDM views.

## [SENSe]:OFDM:SYMBOLs:MAX:STATe

Enables or disables the maximum number of symbols to analyze in the OFDM view.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

```
[SENSe]:OFDM:SYMBOLs:MAX:STATe{OFF|ON|0|1}
```

```
[SENSe]:OFDM:SYMBOLs:MAX:STATe?
```



## Related Commands

[SENSe]:OFDM:SYMBOLs:MAX

## Arguments

OFF or 0 disables the maximum number of symbols to analyze.

ON or 1 enables the maximum number of symbols to analyze.

## Examples

SENSe:OFDM:SYMBOLS:MAX:STATE ON enables the number of symbols to analyze in the OFDM view. The number is specified by the [SENSe]:OFDM:SYMBOLs:MAX command.

## [SENSe]:OFDM:SYMBOLs:RANGe:COUNT

Sets or queries the number of symbols to display in the OFDM view when displaying a range of symbols.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

[SENSe]:OFDM:SYMBOLs:RANGe:COUNT<value>

[SENSe]:OFDM:SYMBOLs:RANGe:COUNT?

## Arguments

<NR1>

## Examples

SENSe:OFDM:SYMBOLS:RANGE:COUNT 25 sets the number of symbols to display to 25.

## [SENSe]:OFDM:SYMBOLs:RANGe:START

Sets or queries the symbols start range in the OFDM view when displaying a range of symbols.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

[SENSe]:OFDM:SYMBOLs:RANGe:START <value>

[SENSe]:OFDM:SYMBOLs:RANGe:START?

## Arguments

<NR1>

## Examples

SENSe:OFDM:SYMBOLS:RANGE:START 0 sets the start range to 0.

## [SENSe]:OFDM:SYMBOLs:SINGLE:INDEX

Sets or queries the symbol index in the OFDM view when displaying a single symbol.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

[SENSe]:OFDM:SYMBOLs:SINGLE:INDEX <value>

[SENSe]:OFDM:SYMBOLs:SINGLE:INDEX?

## Arguments

<NR1>

## Examples

SENSe:OFDM:SYMBOLS:SINGLE:INDEX 0 sets the index to 0.

## [SENSe]:OFDM:UNIT:FREQuency (No Query Form)

Specifies the frequency units for the OFDM measurement.

## Conditions

Measurement views: OFDM

## Group

Sense commands

## Syntax

[SENSe]:OFDM:UNIT:FREQuency{FREQuency|SCARrier}

## Arguments

FREQuency sets the frequency units to frequency.

SCARrier sets the frequency units to subcarriers.

## Examples

SENSe:OFDM:UNIT:FREQUENCY FREQ sets the frequency units to frequency.

## [SENSe]:OFDM:UNIT:TIME (No Query Form)

Specifies the time units for the OFDM measurement.

### Conditions

Measurement views: OFDM

### Group

Sense commands

### Syntax

```
[SENSe]:OFDM:UNIT:TIME{SEConds|SYMBols}
```

### Arguments

SEConds sets the time units to seconds.

SYMBols sets the time units to symbols.

## Examples

SENSe:OFDM:UNIT:TIME SEC sets the time units to seconds.

## [SENSe]:P25:ANALysis:OFFSet

Sets or queries the Analysis Offset value from the P25 displays.

### Conditions

Measurement views: P25 displays

### Group

Sense commands

### Syntax

```
[SENSe]:P25:ANALysis:OFFSet <value>
```

### Arguments

<NRf>

## Examples

SENSe:P25:ANALYSIS:OFFSET 2 sets the Analysis Offset value to 2.

## [SENSe]:P25:ANALysis:OFFSet:AUTO

Sets or queries the Analysis Offset auto checkbox selection from P25 display.

### Conditions

Measurement views: P25 displays

### Group

Sense commands

### Syntax

```
[SENSe]:P25:ANALysis:OFFSet:AUTO <value>
```

### Arguments

0 unchecks the checkbox.

1 checks the checkbox.

### Returns

0 or 1, <NRf>

### Examples

`SENSe:P25:ANALYSIS:OFFSET:AUTO 1` enables the Analysis Offset auto checkbox in P25 display.

## [SENSe]:P25([:BANDwidth]:BWIDth):MODE

Sets or queries the Measurement BW (bandwidth) (frequency span) mode as Auto, Manual, or Link to Span.

### Conditions

Measurement view: Any P25 measurement display

### Group

Sense commands

### Syntax

```
[SENSe]:P25([:BANDwidth]:BWIDth):MODE{MANual|AUTO|LSP}
```

```
[SENSe]:P25([:BANDwidth]:BWIDth):MODE?
```

### Arguments

`MANual` sets the analyzer to allow for manual selection of the measurement bandwidth. A typical use is setting a narrow measurement bandwidth on a small range of frequencies to examine one signal while rejecting the others that are present in a Spectrum graph.

`AUTO` sets analyzer to automatically select a measurement bandwidth. The measurement picks a bandwidth based on other parameter settings (such as symbol rate and modulation type).

`LSP` sets the Measurement BW to Link to Span. This allows you to use the Spectrum display to tune the frequency, view the signal, and then use the Span control to set the measurement bandwidth. This mode emulates legacy instruments.

## Returns

MAN means the Measurement BW is set to Manual.

AUTO means the Measurement BW is set to Automatic.

LSP means the Measurement BW is set to Link to Span.

## Examples

`SENSe:P25:BAND:MODE?` might return MAN, indicating the Measurement BW mode is set to AUTO.

## [SENSe]:P25([:BANDwidth]:BWIDth):TINterval

Sets or queries the Measurement BW (bandwidth) (frequency span). Setting the Measurement BW changes this setting from Auto to Manual if it is set to Auto before you use the command. A query will not change that setting.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

```
[SENSe]:P25([:BANDwidth]:BWIDth):TINterval <NRf>
```

```
[SENSe]:P25([:BANDwidth]:BWIDth):TINterval?
```

## Arguments

<NRf> specifies the measurement bandwidth in kHz.

## Returns

<NR3> = a floating point value with an exponent, which is the Measurement BW value in kHz.

## Examples

`SENSe:P25:BAND:INT?` might return 153.6000000000E+3, which indicates the Measurement BW value is 153.6 kHz.

## [SENSe]:P25([:BANDwidth]:BWIDth):TINterval:AUTO

Sets or queries the Measurement BW (bandwidth) (frequency span) mode as Auto or Manual.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

```
[SENSe]:P25([:BANDwidth]|:BWIDth):TINterval:AUTO{0|1|OFF|ON}  
[SENSe]:P25([:BANDwidth]|:BWIDth):TINterval:AUTO?
```

## Arguments

OFF or 0 sets Measurement BW to Auto.

ON or 1 sets Measurement BW to Manual.

## Returns

0 means Measurement BW is set to Auto.

1 means Measurement BW is set to Manual.

## Examples

SENSe:P25:BAND:INT:AUTO? might return 1, indicating the Measurement BW is set to Auto.

## [SENSe]:P25:FILTer:MEASurement

Sets or queries the P25 measurements Measurement Filter setting. The User Meas Filter selections are predefined user filters.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

```
[SENSe]:P25:FILTer:MEASurement{USER1|USER2|OFF|UOTH|USER3|RECT|GAUS|RRC|OFF|RCOS|  
HDQPSK_P25|C4FM_P25}  
[SENSe]:P25:FILTer:MEASurement?
```

## Arguments

OFF sets the filter to None.

USER1 sets the filter to User Meas Filter 1.

USER2 sets the filter to User Meas Filter 2.

USER3 sets the filter to User Meas Filter 3.

UOTH sets the filter to User other.

RECT sets the filter to Rectangular (only available for Phase 2 HDQPSK modulation type).

RRC sets the filter to Root Raised Cosine (only available for Phase 2 HDQPSK modulation type).

RCOS sets the filter to Raised Cosine (only available for Phase 2 HDQPSK modulation type).

GAUS sets the filter to Gaussian (only available for Phase 2 HDQPSK modulation type).

HDQPSK\_P25 sets the filter to HDQPSK-P25 (only available for Phase 2 HDQPSK modulation type)

C4FM\_P25 sets the filter to C4FM-P25 (only available for Phase 1 C4FM modulation type)

## Returns

OFF means the filter is set to None.

USER1 means the filter is set to User Meas Filter 1.

USER2 means the filter is set to User Meas Filter 2.

USER3 means the filter is set to User Meas Filter 3.

UOTH means the filter is set to User other.

RECT means the filter is set to Rectangular (only available for Phase 2 HDQPSK modulation type).

RRC means the filter is set to Root Raised Cosine (only available for Phase 2 HDQPSK modulation type).

RCOS means the filter is set to Raised Cosine (only available for Phase 2 HDQPSK modulation type).

GAUS means the filter is set to Gaussian (only available for Phase 2 HDQPSK modulation type).

HDQPSK\_P25 means the filter is set to HDQPSK-P25 (only available for Phase 2 HDQPSK modulation type)

C4FM\_P25 means the filter is set to C4FM-P25 (only available for Phase 1 C4FM modulation type)

## Examples

`SENSe:P25:FILTer:MEASUREMENT?` might return USER1, indicating that the Measurement Filter is set to User Meas Filter 1.

## [SENSe]:P25:FILTer:REference

Sets or queries the P25 measurements Reference Filter setting. The User Meas Filter selections are predefined user filters.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

```
[SENSe]:P25:FILTer:REference{USER1|USER2|OFF|UOTH|USER3|RECT|GAUS|RRC|OFF|RCOS|HDQPSK_P25|C4FM_P25}
```

```
[SENSe]:P25:FILTer:REference?
```

## Arguments

OFF sets the filter to None.

USER1 sets the filter to User Meas Filter 1.

USER2 sets the filter to User Meas Filter 2.

USER3 sets the filter to User Meas Filter 3.

UOTH sets the filter to User other.

RECT sets the filter to Rectangular (only available for Phase 2 HDQPSK modulation type).

RCOS sets the filter to Raised Cosine (only available for Phase 2 HDQPSK modulation type).

GAUS sets the filter to Gaussian (only available for Phase 2 HDQPSK modulation type).

## Returns

OFF means the filter is set to None.

USER1 means the filter is set to User Meas Filter 1.

USER2 means the filter is set to User Meas Filter 2.

USER3 means the filter is set to User Meas Filter 3. UOTH means the filter is set to User other.

RECT means the filter is set to Rectangular (only available for Phase 2 HDQPSK modulation type).

RCOS means the filter is set to Raised Cosine (only available for Phase 2 HDQPSK modulation type).

GAUS means the filter is set to Gaussian (only available for Phase 2 HDQPSK modulation type).

## Examples

`SENSe:P25:FILTER:REFERENCE?` might return USER1, indicating that the Reference Filter is set to User Meas Filter 1.

## [SENSe]:P25:FREQuency:ERRor

Sets or queries the P25 Frequency Offset (when Auto is off).

## Conditions

Measurement view: Any P25 measurement display Frequency Offset set to manual (Auto off).

## Group

Sense commands

## Syntax

`[SENSe]:P25:FREQuency:ERRor <NR2>`

`[SENSe]:P25:FREQuency:ERRor?`

## Related Commands

## Arguments

<NR2> = a floating point value without an exponent, which is the fixed frequency offset in Hz.

## Returns

<NR2> = a floating point value without an exponent, which is the fixed frequency offset in Hz.

## Examples

`SENSe:P25:FREQUENCY:ERROR?` might return 749.9999390000, indicating that the frequency offset is 750.0 Hz.

`SENSe:P25:FREQUENCY:ERROR 700.99` sets the frequency offset the 701.0 Hz.



## [SENSe]:P25:FREQuency:ERRor:AUTO

Sets or queries if the P25 Frequency Error is set to Auto.

### Conditions

Measurement view: Any P25 measurement display

### Group

Sense commands

### Syntax

```
[SENSe]:P25:FREQuency:ERRor:AUTO { 0 | 1 | OFF | ON }
```

```
[SENSe]:P25:FREQuency:ERRor:AUTO?
```

### Related Commands

### Arguments

OFF or 0 means the Auto is off.

ON or 1 means the Auto is on.

### Returns

0 means that Auto is off.

1 means Auto is on.

### Examples

`SENSe:P25:FREQUENCY:ERROR:AUTO?` might return 1, indicating that the Frequency Error is set to Auto.

`SENSe:P25:FREQUENCY:ERROR:AUTO 0` turns off Auto setting for Frequency Error.

## [SENSe]:P25:MODUlation:CHANnel:([BANDwidth]:BWIDth)? (Query Only)

Queries the actual P25 Preset channel bandwidth.

### Conditions

Measurement view: Any P25 measurement display

### Group

Sense commands

### Syntax

```
[SENSe]:P25:MODUlation:CHANnel:([BANDwidth]:BWIDth)?
```

### Returns

Bandwidth in kHz, using ‘\_’ to denote “.”. For example, 12\_5 means 12.5 kHz.

## Examples

`SENSe:P25:MODU:CHAN:BAND?` will return `12_5`, indicating that the channel bandwidth is 12.5 kHz.

## [SENSe]:P25:MODUlation:STANdard

Sets or queries the P25 modulation standard.

### Conditions

Measurement view: Any P25 measurement display

### Group

Sense commands

### Syntax

```
[SENSe]:P25:MODUlation:STANdard{PHASE1|PHASE2}
```

```
[SENSe]:P25:MODUlation:STANdard?
```

### Related Commands

`SENSe:P25:MODUlation:TYPE`

### Arguments

PHASE1 is the Phase 1 standard.

PHASE2 is the Phase 2 standard.

### Returns

PHASE1 is the Phase 1 standard.

PHASE2 is the Phase 2 standard.

## Examples

`SENSe:P25:MODULATION:STANDARD?` might return `PHASE2`, indicating that the P25 modulation standard is set to Phase 2.

## SENSe:P25:MODUlation:TYPE

Sets or queries the P25 modulation type.

### Conditions

Measurement view: Any P25 measurement display

### Group

Sense commands

### Syntax

```
SENSe:P25:MODUlation:TYPE{C4FM|HCPM|HDQPSK}
```

SENSe:P25:MODulation:TYPE?

## Related Commands

[SENSe]:P25:MODulation:STANdard

## Arguments

C4FM: this is a Phase 1 modulation standard type. Modulation Phase must be set to PHASE1.

HCPM: this is a Phase 2 modulation standard type. Modulation Phase must be set to PHASE2.

HDQPSK: this is a Phase 2 modulation standard type. Modulation Phase must be set to PHASE2.

## Returns

C4FM: this is a Phase 1 modulation standard type.

HCPM: this is a Phase 2 modulation standard type.

HDQPSK: this is a Phase 2 modulation standard type.

## Examples

SENSe:P25:MODULATION:TYPE? might return HDQPSK, indicating that the P25 modulation type is set to HDQPSK, which is a Phase 2 modulation standard.

## [SENSe]:P25:SRATe

Sets or queries the Symbol Rate value for the P25 measurements.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

[SENSe]:P25:SRATe <value>

[SENSe]:P25:SRATe?

## Arguments

<value> is the symbol rate value in kHz.

## Returns

<NR3> = a floating point value with an exponent, which is the symbol rate value in kHz.

## Examples

SENSe:P25:SRATE? might return 6.000000000E+3, indicating that the symbol rate is 6 kHz.

## **[SENSe]:P25:SUMMARY:CLEar:RESults (No Query Form)**

Clears the results of the P25 Summary display.

### **Conditions**

Measurement view: P25 Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:SUMMARY:CLEar:RESults
```

### **Arguments**

None

### **Examples**

`SENSe:P25:SUMMARY:CLEAR:RESULTS` clears the results in the P25 Summary display.

## **[SENSe]:P25:SUMMARY:LIMIt:FA**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Operating Frequency Accuracy. The limit comparison can be enabled (turned on) or disabled (turned off).

### **Conditions**

Measurement view: P25 Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:SUMMARY:LIMIt:FA <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIt:FA?
```

### **Arguments**

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 4

### **Returns**

<NR1>,<NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FA?` might return "1,2" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for Operating Frequency Accuracy is 2.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ONE

Sets or queries for the On/Off state of limit comparison and the upper limit value for Average t\_error\_1. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ONE <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ONE?
```

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 25

### Returns

<NR1>,<NR3>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ONE?` might return "0,25.0000000000E-6" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Average t\_error\_1 is <25.0000000000E-6>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ZERO

Sets or queries for the On/Off state of limit comparison and the upper limit value for Average t\_error\_0. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ZERO <NR1>,<NR2>
```

[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ZERO?

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 25

### Returns

<NR1>, <NR3>

### Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:AVERAGE:TERROR:ZERO? might return "0,25.0000000000E-6" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Average t\_error\_0 is <25.0000000000E-6>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL:OFF:SLOT

Sets or queries for the On/Off state of limit comparison and the upper limit value for HCPM Transmitter Logic channel offslot. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL: OFF:SLOT <NR1>, <NR2>

[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL: OFF:SLOT?

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to -57

### Returns

<NR1>, <NR2>

### Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL:OFF:SLOT? might return "0,-57" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for HCPM Transmitter Logic channel offslot is <-57>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL:PEAK:ACPR

Sets or queries for the On/Off state of limit comparison and the upper limit value for HCPM Transmitter Logic channel Peak ACPR. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL: PEAK:ACPR <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL: PEAK:ACPR?
```

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 35

### Returns

<NR1>,<NR2>

### Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:LOGIC:CHANNEL:PEAK:ACPR? might return "0,35" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for HCPM Transmitter Logic channel Peak ACPR is <35>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:LO

Sets or queries for the On/Off state of limit comparison and the lower limit value for Frequency Deviation, negative peak. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE: LO <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE: LO?
```

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 2544

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:LO?` might return "1,2544" <1> indicates measurement is enabled (turned on) for limit comparison, and the lower limit for Frequency Deviation, negative peak, is <2544>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:UP**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Frequency Deviation, negative peak. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:UP <NR1>,<NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:UP?`

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 3111

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:NEGATIVE:UP?` might return "1,3111" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for Frequency Deviation, negative peak, is <3111>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:POSITIVE:LO**

Sets or queries for the On/Off state of limit comparison and the lower limit value for Frequency Deviation, positive peak. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary



## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PEAK:POSItive: LO <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PEAK:POSItive: LO?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 2544

## Returns

<NR1>,<NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:POSITIVE:LO?` might return "1,2544" <1>indicates measurement is enabled (turned on) for limit comparison, and the lower limit for Frequency Deviation, positive peak, is <2544>.

## [SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PEAK:POSItive:UP

Sets or queries for the On/Off state of limit comparison and the upper limit value for Frequency Deviation, positive peak. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PEAK:POSItive: UP <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PEAK:POSItive: UP?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 3111 Hz

## Returns

<NR1>,<NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PEAK:POSITIVE:UP?` might return "1,3111" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for Frequency Deviation, positive peak, is <3111>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO**

Sets or queries for the On/Off state of limit comparison and the lower limit value for Frequency Deviation, f1/f2/f3/f4 for Phase1 or Phase2. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO <NR1>, <NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO?`

## Arguments

<NR1> =0 to deselect the measurement and 1 to select the measurement.

<NR2> =0 to 2544

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO?` might return "1,2544" <1> indicating measurement is enabled (turned on) for limit comparison, and the upper limit for Frequency Deviation is <2544>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO**

Sets or queries for two values namely, the On/Off state of limit comparison and the lower limit value for Frequency deviation, f1/f2/f3/f4 for Phase1 or Phase2. The limit comparison can be enabled (turned On) or disabled (turned off).

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:LO <NR1>,<NR2>
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:LO?
```

## Arguments

<Integer, Integer>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:LO? returns "1,2544" Indicating the measurement is enabled for limit comparison and the lower limit for Frequency deviation is <2544>.

## [SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:UP

Sets or queries for the On/Off state of limit comparison and the upper limit value for Frequency Deviation, f1/f2/f3/f4 for Phase1 or Phase2. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:UP <NR1>,<NR2>
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:UP?
```

## Arguments

<NR1> =0 to deselect the measurement and 1 to select the measurement.

<NR2> =0 to 3111

## Returns

<NR1>,<NR2>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:UP? might return "1,3111" <1> indicating measurement is enabled (turned on) for limit comparison, and the upper limit for Frequency Deviation is <3111>.

## [SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PHASe[1/2]:F[1/2/3/4]:UP

Sets or queries for two values namely, the On/Off state of limit comparison and the upper limit value for Frequency deviation, f1/f2/f3/f4 for Phase1 or Phase2. The limit comparison can be enabled (turned On) or disabled ( turned off).

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASe[1/2]: F[1/2/3/4]:UP <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASe[1/2]:F[1/2/3/4]:UP?
```

## Arguments

<Integer, Integer>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PHASE[1/2]:F[1/2/3/4]:UP? returns "1,3111"  
" Indicating the measurement is enabled for limit comparison and the upper limit for Frequency deviation is <3111>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF

Sets or queries for the On/Off state of limit comparison and the upper limit value for Pmax-off. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF <NR1>,<NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF?
```

## Arguments

<NR1> =0 to deselect the measurement and 1 to select the measurement.

<NR2> =0 to 4

## Returns

<NR1>,<NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF?` might return "0,4" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Pmax-off is <4>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF**

Sets or queries for two values namely, the On/Off state of limit comparison and the upper limit value for Pmax-off. The limit comparison can be enabled (turned On) or disabled (turned off).

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Sense commands

## Syntax

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF<NR1>,<NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF?`

## Arguments

<Integer, Integer>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:OFF?` returns "0,4" <0> indicates Measurement is disabled (turned Off) for limit comparison, and the upper limit for Pmax-off is <4>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Pmax-on. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON <NR1>,<NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON?`

## Arguments

<NR1> =0 to deselect the measurement and 1 to select the measurement.

<NR2> =0 to 4

## Returns

<NR1>, <NR2>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON? might return "0,4" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Pmax-on is <4>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON

Sets or queries for two values namely, the On/Off state of limit comparison and the upper limit value for Pmax-on. The limit comparison can be enabled (turned On) or disabled ( turned off).

## Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

## Group

Sense commands

## Syntax

[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON<NR1>, <NR2>

[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON?

## Arguments

<Integer, Integer>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:ON? returns "0,4" <0> indicates Measurement is disabled (turned Off) for limit comparison, and the upper limit for Pmax-on is <4>.

## [SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:SS

Sets or queries for the On/Off state of limit comparison and the upper limit value for Pss-max. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PMAX:SS<NR1>, <NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PMAX:SS?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 1

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMAX:SS?` might return "0,1" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Pss-max is <1>.

## [SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PMIN:SS

Sets or queries for the On/Off state of limit comparison and the upper limit value for Pss-min. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PMIN:SS<NR1>, <NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:PMIN:SS?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to -3

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:PMIN:SS?` might return "0,-3" <0> indicating measurement is disabled (turned off) for limit comparison, and the upper limit for Pss-min is <-3>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:ENCODER**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack encoder. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:ENCODER <NR1>, <NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:ENCODER?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 100

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER:AVERAGE:ATTK:ENCODER?` might return "0,100" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for Transmitter Average Attack encoder is <100>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:ENCODER:BI**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack encoder, busy/idle. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands



## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:TRANsmitter: AVERage:ATTK:ENCODer:BI
<NR1>, <NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:TRANsmitter: AVERage:ATTK:ENCODer:BI?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 30

## Returns

<NR1>, <NR2>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER:AVERAGE:ATTK:ENCODER:BI? might return "0,30" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for Transmitter Average Attack encoder, busy/idle, is <30>.

## [SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:TRANsmitter: AVERage:ATTK:POWER

Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack power. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:TRANsmitter: AVERage:ATTK:POWER
<NR1>, <NR2>
```

```
[SENSe]:P25:SUMMary:LIMIt:FREQuency:DEVIation:TRANsmitter: AVERage:ATTK:POWER?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 50

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER:AVERAGE:ATTK:POWER?` might return "0,50" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for Transmitter Average Attack power is <50>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:POWER:BI**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average Attack power, busy/idle. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:POWER:BI`  
`<NR1>, <NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:ATTK:POWER:BI?`

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 30

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER:AVERAGE:ATTK:POWER:BI?` might return "0,30" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for Transmitter Average Attack power, busy/idle, is <30>.

## **[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:THRUput:DLAY**

Sets or queries for the On/Off state of limit comparison and the upper limit value for Transmitter Average throughput delay. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:THRUPUT:DLAY <NR1>, <NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER: AVERAGE:THRUPUT:DLAY?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 125

## Returns

<NR1>, <NR2>

## Examples

SENSe:P25:SUMMARY:LIMIT:FREQUENCY:DEVIATION:TRANSMITTER:AVERAGE:THRUPUT:DLAY? might return "0,125" <0> indicates measurement is disabled (turned off) for limit comparison, and the upper limit for Transmitter Average throughput delay is <125>.

## [SENSe]:P25:SUMMARY:LIMIT:MF

Sets or queries for the On/Off state of limit comparison and the upper limit value for Modulation Fidelity. The limit comparison can be enabled (turned on) or disabled (turned off).

## Conditions

Measurement view: P25 Summary

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SUMMARY:LIMIT:MF <NR1>, <NR2>
```

```
[SENSe]:P25:SUMMARY:LIMIT:MF?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 4

## Returns

<NR1>, <NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:MF?` might return "1,4" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for Modulation Fidelity is <4>.

## [SENSe]:P25:SUMMARY:LIMIT:RF:OUTPower

Sets or queries for the On/Off state of limit comparison and the upper limit value for RF Output Power. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

`[SENSe]:P25:SUMMARY:LIMIT:RF:OUTPower <NR1>,<NR2>`

`[SENSe]:P25:SUMMARY:LIMIT:RF:OUTPower?`

### Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 3

### Returns

<NR1>,<NR2>

## Examples

`SENSe:P25:SUMMARY:LIMIT:RF:OUTPOWER?` might return "1,3" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for RF Output Power is <3>.

## [SENSe]:P25:SUMMARY:LIMIT:SRA

Sets or queries for the On/Off state of limit comparison and the upper limit value for Symbol Rate Accuracy. The limit comparison can be enabled (turned on) or disabled (turned off).

### Conditions

Measurement view: P25 Summary

### Group

Sense commands

### Syntax

`[SENSe]:P25:SUMMARY:LIMIT:SRA <NR1>,<NR2>`

```
[SENSe]:P25:SUMMArY:LIMIT:SRA?
```

## Arguments

<NR1> = 0 to deselect the measurement and 1 to select the measurement.

<NR2> = 0 to 10

## Returns

<NR1>, <NR2>

## Examples

SENSe:P25:SUMMARY:LIMIT:SRA? might return "1,10" <1> indicates measurement is enabled (turned on) for limit comparison, and the upper limit for Symbol Rate Accuracy is <10>.

## [SENSe]:P25:SYMBOLS:POINTS

Sets or queries the Points/Symbols to display in the P25 measurement view.

## Conditions

Measurement view: Any P25 measurement display

## Group

Sense commands

## Syntax

```
[SENSe]:P25:SYMBOLS:POINTS{ONE|TWO|FOUR|EIGHT}
```

```
[SENSe]:P25:SYMBOLS:POINTS?
```

## Arguments

ONE set the Points/Symbol trace setting to 1.

TWO set the Points/Symbol trace setting to 2.

FOUR set the Points/Symbol trace setting to 4.

EIGHT set the Points/Symbol trace setting to 8.

## Returns

ONE means the Points/Symbol trace setting is set to 1.

TWO means the Points/Symbol trace setting is set to 2.

FOUR means the Points/Symbol trace setting to is set 4.

EIGH means the Points/Symbol trace setting to is set 8.

## Examples

SENSe:P25:SYMBOLS:POINTS? might return 4, indicating that the Points/Symbol value is set to 4.

## **[SENSe]:P25:TEST:PATtern**

Sets or returns the test pattern type in the P25 displays.

### **Conditions**

Measurement views: Any Bluetooth display

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:TEST:PATtern{LDEVIation|HDEVIation|TPOTHER}
```

```
[SENSe]:P25:TEST:PATtern?
```

### **Arguments**

LDEVIation indicates the low deviation test pattern.

HDEVIation indicates the high deviation test pattern.

TPOTHER indicates a user defined test pattern.

### **Examples**

SENSe:P25:TEST:PATTERN? might return LDEV indicating the selected test pattern type is the low deviation type.

## **[SENSe]:P25:TIME:UNITs**

Sets or queries the fundamental unit of time for the P25 measurements.

### **Conditions**

Measurement view: Any P25 measurement display

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:TIME:UNITs{SEC|SYMB}
```

```
[SENSe]:P25:TIME:UNITs?
```

### **Arguments**

SEConds sets the unit of time to seconds.

SYMBols sets the unit of time to symbols.

### **Returns**

SEC means that the unit if time is set to seconds.

SYMB means that the unit if time is set to symbols.

## Examples

`SENSe:P25:TIME:UNITS?` might return SEC, indicating that the unit of time is set to seconds.

## [SENSe]:P25:TRIGgger:MEASurement

Sets or queries the Trigger Measurement from the P25 displays.

### Conditions

Measurement views: P25 displays

### Group

Sense commands

### Syntax

```
[SENSe]:P25:TRIGgger:MEASurement <arg1>
```

```
[SENSe]:P25:TRIGgger:MEASurement?
```

### Arguments

TATOBSYNC is the Time Alignment (tOB\_sync measurement).

TAERR0 is the Time Alignment (t\_error\_0\_calculation).

TAERR1 is the Time Alignment (t\_error\_1\_calculation).

TIMEASNONE is None.

### Returns

See Arguments

## Examples

`SENSe:P25:TRIGGER:MEASUREMENT TATOBSYNC` selects the Time Alignment (tOB\_sync measurement) option when Modulation Params Standard is Phase2 and Modulation Type is C4FM.

## [SENSe]:P25:TRIGgger:MEASurement:TOBSync

Sets or returns the tOB\_sync (measured) value in the P25 displays.

### Conditions

Measurement views: P25 displays

### Group

Sense commands

## Syntax

[SENSe]:P25:TRIGgger:MEASurement:TOBSync <value>

## Arguments

None

## Examples

SENSe:P25:TRIGGER:MEASUREMENT:TOBSYNC? might return 6.00000000, which is the tOB\_sync(measured) value.

## [SENSe]:P25:TRIGgger:MEASurement:TOBSync:AUtO

Sets or queries the Auto checkbox selection from the P25 displays.

## Conditions

Measurement views: P25 displays

## Group

Sense commands

## Syntax

[SENSe]:P25:TRIGgger:MEASurement:TOBSync:AUtO <value>

## Arguments

0 unchecks the checkbox.

1 checks the checkbox.

## Returns

0 or 1, <NRf>

## Examples

SENSe:P25:TRIGGER:MEASUREMENT:TOBSync:AUtO 1 enables the Auto checkbox in Trig Meas tab.

## [SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour

Sets or queries the Trigger Measurements option Phase1-Transient Frequency Behavior from the P25 displays.

## Conditions

Measurement views: P25 displays

## Group

Sense commands



## Syntax

```
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency: BEHAViour{RISE|FALL}
```

## Arguments

RISE sets the Transient Frequency Behavior to Off to On.

FALL sets the Transient Frequency Behavior to On to Off.

## Returns

RISE means the Transient Frequency Behavior is Off to On.

FALL means the Transient Frequency Behavior is On to Off.

## Examples

`SENSe:P25:TRIGGGER:MEASUREMENT:TRANSIENT:FREQUENCY:BEHAVIOUR?` might return RISE if the Transient Frequency behavior Off to On option is chosen in the Transient Frequency.

## [SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:OFF

Sets or queries the 't off' value when the Transient Frequency option is Transient Frequency Behavior On to Off for Phase1-Transient Frequency Behavior from the P25 displays.

## Conditions

Measurement views: P25 displays

## Group

Sense commands

## Syntax

```
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency: BEHAViour:TIME:OFF <arg1>
```

```
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:OFF?
```

## Arguments

<arg1>

## Returns

<arg1>::<Nrf>

## Examples

`SENSe:P25:TRIGGGER:MEASUREMENT:TRANSIENT:FREQUENCY:BEHAVIOUR:TIME:OFF 2` sets the "t off" value to 2.

## **[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:OFF:AUTO**

Sets or queries the Auto checkbox selection when Transient Frequency is chosen as Transient Frequency Behavior On to Off from the P25 displays.

### **Conditions**

Measurement views: P25 displays

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency: BEHAViour:TIME:OFF:AUTO  
<value>
```

### **Arguments**

0 unchecks the checkbox.

1 checks the checkbox.

### **Returns**

0 or 1, <NRF>

### **Examples**

`SENSe:P25:TRIGGER:MEASUREMENT:TRANSIENT:FREQUENCY:BEHAVIOUR:TIME:OFF:AUTO 1` enables the Auto checkbox in Trig measurement tab.

## **[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:ON**

Sets or queries the 't on' value when the Transient Frequency option is Transient Frequency Behavior Off to On for Phase1-Transient Frequency Behavior from the P25 displays.

### **Conditions**

Measurement views: P25 displays

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:ON <arg1>  
[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAViour:TIME:ON <arg1>
```

## Arguments

<arg1>

## Examples

`SENSe:P25:TRIGGER:MEASUREMENT:TRANSIENT:FREQUENCY:BEHAVIOUR:TIME:ON 2` sets the "t on" value to 2.

## **[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency:BEHAviour:TIME:ON:AUTO**

Sets or queries the Auto checkbox selection when Transient Frequency is chosen as Transient Frequency behavior Off to On from the P25 displays.

## Conditions

Measurement views: P25 displays

## Group

Sense commands

## Syntax

`[SENSe]:P25:TRIGgger:MEASurement:TRANSient:FREQuency: BEHAviour:TIME:ON:AUTO`  
<value>

## Arguments

0 unchecks the checkbox.

1 checks the checkbox.

## Returns

0 or 1, <NRf>

## Examples

`SENSe:P25:TRIGGER:MEASUREMENT:TRANSIENT:FREQUENCY:BEHAVIOUR:TIME:ON:AUTO 1` enables the Auto checkbox in Trig measurement tab.

## **[SENSe]:PERRor[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source of Phase error versus Time for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Sense commands

## Syntax

```
[SENSE]:PERRor[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSE]:PERRor[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

SENSE:PERRor:MEASview1:SELEcted:SOURce? queries the Source for Phase error versus Time Measurement View 1.

## [SENSe]:PHVTime[:MEASview<y>]:CLEAr:RESuLts (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:CLEAr:RESuLts
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

SENSe:PHVTIME:MEASview1:CLEAr:RESuLTS restarts multi-trace functions.

## [SENSe]:PHVTime[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the Phase versus Time measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:CENTER <value>
```

```
[SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:CENTER?
```

## Related Commands

```
[SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:START,
```

```
[SENSE]:PHVTime:MEASview<y>:FREQUENCY:STOP
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

SENSE:PHVTIME:MEASview1:FREQUENCY:CENTER 7.5GHz sets the center frequency to 7.5 GHz.

## [SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN

Sets or queries the frequency span in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN <value>
```

```
[SENSE]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value> : : <NRf> is the frequency span.

Range: 10 Hz to 40 MHz (Standard) / 110 MHz (Option 110)

## Examples

SENSe:PHVTime:MEASview1:FREQUENCY:SPAN 20MHz sets the span to 20 MHz.

## [SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN:LINK

Sets or queries the frequency span link in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN:LINK <value>
```

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:SPAN:LINK?
```

## Arguments

OFF or 0 specifies that Link to Span is not checked.

ON or 1 specifies that Link to Span is checked .

## Examples

SENSe:PHVTime:MEASview2:FREQUENCY:SPAN:LINK? Might return 0 or 1 based on condition of Link to Span checkbox.

## [SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:START

Sets or queries the measurement start frequency (left edge on the graph) in the Phase versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the

[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:CENTer command.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:START <value>
```

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:START?
```

## Related Commands

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STOP
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> is the measurement start frequency.

Range: (center frequency) ± (span)/2.

## Examples

SENSe:PHVTIME:MEASview1:FREQUENCY:START 6.95GHz sets the start frequency to 6.95 GHz.

## [SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO OFF.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP <value>
```

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP?
```

## Related Commands

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz

- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:PHVTIME:MEASview1:FREQUENCY:STEP 1.5kHz` sets the step size to 1.5 kHz.

## **[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO**

Determines whether to set the frequency step size automatically or manually.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP:AUTO?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 specifies that the frequency step size is set manually using the `[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STEP` command.

ON or 1 specifies that the frequency step size is set automatically.

## Examples

`SENSe:PHVTIME:MEASview1:BANDWIDTH:AUTO ON` sets the frequency step size automatically.

## **[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STOP**

Sets or queries the measurement stop frequency (right edge of the graph) in the Phase versus Time measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:CENTer` command.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:FREQUENCY:STOP <value>
```



```
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:STOP?
```

## Related Commands

```
[SENSe]:PHVTime[:MEASview<y>]:FREQuency:START
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<value>::<NRF> is the measurement stop frequency.

Range: (center frequency) ± (span)/2.

## Examples

```
SENSe:PHVTIME:MEASview1:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.
```

## [SENSe]:PHVTime[:MEASview<y>]:MAXTracepoints

Sets or queries the maximum trace points in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Sense commands

## Syntax

```
[SENSe]:PHVTime[:MEASview<y>]:MAXTracepoints { ONEK | TENK | HUNDredk | NEVerdecimate }
```

```
[SENSe]:PHVTime[:MEASview<y>]:MAXTracepoints?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

ONEK sets the maximum trace points to 1 k.

TENK sets the maximum trace points to 10 k.

HUNDredk sets the maximum trace points to 100 k.

NEVerdecimate never decimates the trace points.

## Examples

```
SENSe:PHVTIME:MEASview1:MAXTRACEPOINTS TENK sets the maximum trace points to 10 k.
```

## **[SENSe]:PHVTime[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source with respect to the Phaseversus Time Measurement View number

### **Conditions**

Measurement views: Phase versus Time with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### **Group**

Sense commands

### **Syntax**

```
SENSe:PHVTime[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"  
| "Source 4"}
```

```
SENSe: PHVTime[:MEASview<y>]:SELEcted:SOURce?
```

### **Arguments**

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### **Examples**

SENSe:PHVTime:MEASview1:SELEcted:SOURce? queries or sets the Source for Phase versus Time Measurement View 1.

## **[SENSe]:PM:PHASe:OFFSet**

Sets or queries the phase offset in the PM measurement.

### **Conditions**

Measurement views: PM

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PM:PHASe:OFFSet <value>
```

```
[SENSe]:PM:PHASe:OFFSet?
```

### **Arguments**

<value>::<NRf> specifies the phase offset. Range: -180 to +180 °.

### **Examples**

SENSe:PM:PHASe:OFFSet 10deg sets the phase offset to 10 °.

## [SENSe]:PM:PHASe:OFFSet:MARKer (No Query Form)

Sets the phase offset from the selected marker location in the PM measurement.

### Conditions

Measurement views: PM

### Group

Sense commands

### Syntax

```
[SENSe]:PM:PHASe:OFFSet:MARKer
```

### Arguments

None.

### Examples

`SENSe:PM:PHASe:OFFSet:MARKer` sets the phase offset from the selected Marker position.

## [SENSe]:PM:PHASe:SEARch:AUTO

Determines whether to detect the phase automatically or manually in the PM measurement.

### Conditions

Measurement views: PM

### Group

Sense commands

### Syntax

```
[SENSe]:PM:PHASe:SEARch:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:PM:PHASe:SEARch:AUTO?
```

### Arguments

OFF or 0 specifies that the phase offset is set manually using the command `[SENSe]:PM:PHASe:OFFSet`.

ON or 1 specifies that the phase is detected automatically.

### Examples

`SENSe:PM:PHASe:SEARch:AUTO ON` specifies that the phase is detected automatically.

## [SENSe]:PNOise:AVERage:COUNT

Sets or queries the number of traces to combine for averaging in the phase noise measurement. This command is effective when

`[SENSe]:PNOise:AVERage:ENABle` is set to ON.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSE]:PNOise:AVERage:COUNT <number>
```

```
[SENSe]:PNOise:AVERage:COUNT?
```

## Arguments

<number>: : <NR1> specifies the average count. Range: 2 to 10000.

## Examples

SENSe: PNOISE: AVERAGE: COUNT 64 sets the average count to 64.

## [SENSe]:PNOise:AVERage:ENABLE

Determines whether to enable averaging trace in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSE]:PNOise:AVERage:ENABLE{OFF|ON|0|1}
```

```
[SENSe]:PNOise:AVERage:ENABLE?
```

## Arguments

OFF disables averaging trace.

ON enables averaging trace.

## Examples

SENSe: PNOISE: AVERAGE: ENABLE ON enables averaging trace.

## [SENSe]:PNOise:CARRier:FREQuency:TRACK

Determines whether to enable or disable tracking the carrier frequency in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:CARRier:FREQuency:TRACk{OFF|ON|0|1}
```

```
[SENSe]:PNOise:CARRier:FREQuency:TRACk?
```

## Arguments

OFF or 0 disables tracking the carrier frequency.

ON or 1 enables tracking the carrier frequency.

## Examples

SENSe:PNOISE:CARRIER:FREQUENCY:TRACK ON enables tracking the carrier frequency.

## [SENSe]:PNOise:CARRier:THReshold

Sets or queries the threshold level to detect the carrier in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:CARRier:THReshold <value>
```

```
[SENSe]:PNOise:CARRier:THReshold?
```

## Arguments

<value>::<NRF> specifies the threshold level above which the input signal is determined to be a carrier. Range: -60 to 0 dBm.

## Examples

SENSe:PNOISE:CARRIER:THRESHOLD -25 sets the carrier threshold level to -25 dB.

## [SENSe]:PNOise:CLEar:RESults (No Query Form)

Restarts the average process, clearing average data and counter.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:CLEar:RESults
```

## Arguments

None

## Examples

SENSe:PNOISE:CLEAR:RESULTS restarts the average process.

## [SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STARt

Sets or queries the start offset frequency for integration in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STARt <value>
```

```
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STARt?
```

## Arguments

<value>::<Nrf> specifies the start offset frequency for integration.

It must be less than the stop offset frequency. Range: 10 Hz to Stop Offset Frequency - 1 Hz.

## Examples

SENSe:PNOISE:FREQUENCY:INTEGRATION:OFFSET:START 100kHz sets the start offset frequency for integration to 100 kHz.

## [SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STOP

Sets or queries the stop offset frequency for integration in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STOP <value>
```

```
[SENSe]:PNOise:FREQuency:INTEgration:OFFSet:STOP?
```

## Arguments

<value> : : <Nrf> specifies the stop offset frequency for integration.

It must be greater than the start offset frequency. Range: 11 Hz to 1 GHz.

## Examples

SENSe:PNOISE:FREQUENCY:INTEGRATION:OFFSET:STOP 100MHz sets the stop offset frequency for integration to 100 MHz.

## [SENSe]:PNOise:FREQuency:PLOT:OFFSet:START

Sets or queries the start offset frequency for plotting the phase noise trace.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:START <value>
```

```
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:START?
```

## Arguments

<value> : : <Nrf> specifies the start offset frequency for the trace plot.

It must be less than the stop offset frequency. Range: 10 Hz to 100 MHz in a tenfold sequence.

## Examples

SENSe:PNOISE:FREQUENCY:PLOT:OFFSET:START 100kHz sets the start offset frequency for plot to 100 kHz.

## [SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP

Sets or queries the stop offset frequency for plotting the phase noise trace.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

```
[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP <value>
```

[SENSe]:PNOise:FREQuency:PLOT:OFFSet:STOP?

## Arguments

<value>::<Nrf> specifies the stop offset frequency for plot.

It must be greater than the start offset frequency. Range: 100 Hz to 1 GHz in a tenfold sequence.

## Examples

SENSe:PNOISE:FREQUENCY:PLOT:OFFSET:STOP 100MHz sets the stop offset frequency for plot to 100 MHz.

## [SENSe]:PNOise:OPTimization

Sets or queries the method of optimizing the gain and input bandwidth in the phase noise measurement.

## Conditions

Measurement views: Phase noise

## Group

Sense commands

## Syntax

[SENSe]:PNOise:OPTimization { DRANge | SPEed }

[SENSe]:PNOise:OPTimization?

## Arguments

DRANge optimizes the gain and input bandwidth to maximize the dynamic range.

SPEed optimizes the gain and input bandwidth to speed the measurement.

## Examples

SENSe:PNOISE:OPTIMIZATION DRANge optimizes the gain and input bandwidth to maximize the dynamic range.

## [SENSe]:POWER:UNITs

Sets or queries the fundamental unit of power.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:POWER:UNITs { DBM | DBV | VOLTs | WATTs | DBUW | DBW | DBUV | DBMV | DBUA | DBUV\_M | DBUA\_M | AMPS }



[SENSe]:POWer:UNITs

## Arguments

The following table lists the arguments.

**Table 52: Power units**

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts
DBUW	dB $\mu$ W
DBW	dBW
DBUV	dB $\mu$ V
DBMV	dBmV
DBUA	dB $\mu$ A
DBUV_M	dB $\mu$ V/m
DBUA_M	dB $\mu$ A/m
AMPS	Amps



**Note:** Select dB $\mu$ V/m or dB $\mu$ A/m unit when using an antenna table.

## Examples

SENSe:POWer:UNITs DBM specifies the fundamental unit of power as dBm.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:ALENgt h

Sets or queries the absolute amount of time included in the Frequency-Domain Linearity type measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:ALENgt h <value>

[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:ALENgt h?

## Arguments

<value> : : <NRf> specifies the absolute time in seconds for the measurement time.

Range: 1 ns to 100 seconds

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

[SENSE]:PULSE:MEASview2:ANALYZE:FDOMAIN:MEASUREMENT:TIME:ALENGTH 1E-6 sets the measurement time to 1  $\mu$ s.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:METHOD

Sets or queries the measurement time method for the Frequency-Domain Linearity type measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSE]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:METHOD { ABSolute | RELative }
```

```
[SENSE]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:METHOD?
```

## Arguments

ABSolute specifies the absolute time in seconds based on rising edge of the pulse.

RELative specifies the relative time in percent based on the pulse width and the center of the pulse.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:ANALyze:FDOMain:MEASurement:TIME:METHOD REL sets the time method to Relative.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLENgth**

Sets or queries the relative amount of time included in the Frequency-Domain Linearity type measurements.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLENgth <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLENgth?
```

### **Arguments**

<value> : :<NRf> specifies the relative time in percent for the measurement time based on the pulse width and centered of the pulse.

Range: 10% to 100%

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf>

### **Examples**

SENSe:PULSe:MEASview2:ANALyze:FDOMain:MEASurement:TIME:RLENgth 70 sets the measurement time to 70% of the pulse width.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLEVel**

Sets or queries the measurement reference level for Frequency-Domain Linearity measurements.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLEVel <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:RLEVel?
```

## Arguments

<value> : : <Nrf> specifies the reference level in percentage as the location on the voltage straight-line approximation of the pulse rising edge.

Range: 50% to 100%

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<Nrf>

## Examples

SENSe:PULSe:MEASview2:ANALyze:FDOMain:MEASurement:TIME:RLEVel 60 sets the reference level to 60% of the rising edge.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:START

Sets or queries the start time for the Frequency-Domain Linearity type measurements. The start time is the amount of time wait after the rising-edge reference before starting to measure.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:START <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:FDOMain:MEASurement:TIME:START?
```

## Arguments

<value> : : <Nrf> specifies the start time in seconds from the reference level of the rising edge of the pulse.

Range: -100 ms to 100 ms

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<Nrf>

## Examples

SENSe:PULSe:MEASview2:ANALyze:FDOMain:MEASurement:TIME:START 1E-6 sets the start time to 1  $\mu$ s.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:CORRection:AMPLitude[:STATe]**

Enables or disables the corrections for window and for delay of the highest side lobe in the Pulse Impulse Response measurements.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:CORRection:AMPLitude[:STATe] { OFF | ON | 0 | 1 }
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:CORRection:AMPLitude[:STATe]?
```

### **Arguments**

OFF or 0 disables amplitude correction in Impulse Response measurements.

ON or 1 enables amplitude correction in Impulse Response measurements.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf>

### **Examples**

SENSe:PULSe:MEASview2:ANALyze:IRESpOse:CORRection:AMPLitude:STATe 1 enables amplitude correction.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:KOTime**

Sets or queries the Keep-out time in the Pulse Impulse Response measurements. This specifies the width of time around the main lobe that lobes within this time zone are not eligible to be the "highest side lobe."

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:KOTime <value>
```

[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:KOTime?

### Arguments

<value>: :<NRf> specifies the Keep-out time in seconds.

Range: 0 to 100 seconds.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

### Examples

SENSe:PULSe:MEASview2:ANALyze:IRESpOse:KOTime 0.001 sets the keep-out time to 1 ms.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:ALENgtH

Sets or queries the absolute amount of time included in the Impulse Response type measurements.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:ALENgtH <value>

[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:ALENgtH?

### Arguments

<value>: :<NRf> specifies the absolute time in seconds for the measurement time.

Range: 1 ns to 100 seconds.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

### Examples

SENSe:PULSe:MEASview2:ANALyze:IRESpOse:MEASurement:TIME:ALENgtH 1E-6 sets the measurement time to 1  $\mu$ s.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:METhod**

Sets or queries the measurement time method for the Impulse Response type measurements.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:METhod { ABSolute | RELative }
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:METhod?
```

### **Arguments**

**ABSolute** specifies the absolute time in seconds based on rising edge of the pulse.

**RELative** specifies the relative time in percent based on the pulse width and the center of the pulse.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf>

### **Examples**

`SENSE:PULSE:MEASview2:ANALYZE:IRESPONSE:MEASUREMENT:TIME:METHOD REL` sets the time method to Relative.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLENgth**

Sets or queries the relative amount of time included in the Impulse Response type measurements.

### **Conditions**

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLENgth <value>
```

[SENSE]:PULSE[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLENgth?

## Arguments

<value> : :<NRf> specifies the relative time in percent for the measurement time based on the pulse width and centered of the pulse.

Range: 10% to 100%.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSE:PULSE:MEASview2:ANALYZE:IRESPONSE:MEASUREMENT:TIME:RLENGTH 70 sets the measurement time to 70% of the pulse width.

## [SENSE]:PULSE[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLEVel

Sets or queries the measurement reference level for Impulse Response measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

[SENSE]:PULSE[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLEVel <value>

[SENSE]:PULSE[:MEASview<y>]:ANALyze:IRESpOse:MEASurement:TIME:RLEVel?

## Arguments

<value> : :<NRf> specifies the reference level in percentage as the location on the voltage straight-line approximation of the pulse rising edge.

Range: 50% to 100%.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>



## Examples

`SENSE:PULSE:MEASview2:ANALYZE:IRESPONSE:MEASUREMENT:TIME:RLEVEL 60` sets the reference level to 60% of the rising edge.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESponse:MEASurement:TIME:STARt**

Sets or queries the start time for the Impulse Response type measurements. The start time is the amount of time wait after the rising-edge reference before starting to measure.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESponse:MEASurement:TIME:STARt <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:IRESponse:MEASurement:TIME:STARt?
```

## Arguments

`<value>::<NRf>` specifies the start time in seconds from the reference level of the rising edge of the pulse.

Range: -100 ms to 100 ms.

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

`<NRf>`

## Examples

`SENSE:PULSE:MEASview2:ANALYZE:IRESPONSE:MEASUREMENT:TIME:START 1E-6` sets the start time to 1  $\mu$ s.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel**

Sets or queries how to determine the 50% level for the pulsed RF measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel{ VOLTage | POWer }  
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel?
```

## Arguments

VOLTage uses -6 dB to determine the 50% level.

POWer uses -3 dB to determine the 50% level.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:ANALyze:LEVel POWer sets RF measurements to use -3 dB to determine the 50% level.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:FIFTy

Sets or queries how to determine the 50% level for the pulsed RF measurements. This command is equivalent to the [SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel command.

## Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:FIFTy { VOLTage | POWer }  
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:FIFTy?
```

## Arguments

VOLTage uses -6 dB to determine the 50% level.

POWer uses -3 dB to determine the 50% level.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

SENSe:PULSe:MEASview2:ANALyze:LEVel:FIFTy POWer sets RF measurements to use -3 dB to determine the 50% level.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:HUNDred

Sets or queries how to determine the 100% level in the pulsed RF measurements. This command is equivalent to the [SENSe]:PULSe:ANALyze:LEVel command.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:HUNDred { AVERage | INDEpendent }
[SENSe]:PULSe[:MEASview<y>]:ANALyze:LEVel:HUNDred?
```

### Arguments

AVERage uses the average amplitude calculated for the pulse-on as the 100% reference to measure the rise and fall times.

INDEpendent uses the amplitudes at the beginning and end of the pulse-on as the 100% references to measure the rise and fall times, respectively

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

### Examples

SENSe:PULSe:MEASview2:ANALyze:LEVel:HUNDRED AVERage uses pulse average amplitude to set the 100% level.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:MEASurement:TIME:TYPE

Sets or Queries the Measurement(s) parameter from the Control panel settings of Define tab which is common across the displays Pulse Statistics, Pulse Table, Pulse Trace, Pulse Cumulative Histogram, Pulse Cumulative Statistics.

### Conditions

Measurement views: Pulse Statistics, Pulse Table, Pulse Trace, Pulse Cumulative Histogram, Pulse Cumulative Statistics

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:MEASurement:TIME:TYPE ?
```

## Arguments

`<arg1> :: {FDL | IMPR | FPD | RIPP | OVER}` sets the control panel Measurement(s) parameter of define tab to Freq-domain Linearity or Impulse Response or F P2R Diff, F P2P Diff, Freq Abs, Phase P2R Diff, Phase P2P Diff or Ripple or Overshoot  
`<y> = 1 to 8`; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

`<string>`

## Examples

`SENSe:PULSe:ANALYZE:MEASview1:MEASUREMENT:TIME:TYPE FDL` sets the control panel Measurement(s) parameter under Define tab to Freq-domain Linearity.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:MEASurement:TIME:TYPE**

Sets or queries the Measurement(s) parameter for the Define tab of the Settings control panel, which is common across the Pulse Statistics, Pulse Table, Pulse Trace, Pulse Cumulative Histogram, and Pulse Cumulative Statistics displays.

## Conditions

Measurement views: All Pulsed RF displays (except Pulse-Ogram)

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:MEASurement:TIME:TYPE { FDL | IMPR | FPD | RIPP | OVER }
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:MEASurement:TIME:TYPE ?
```

## Arguments

FDL is Freq-domain Linearity

IMPR is Impulse Response

FPD is F P2R Diff

RIPP is ripple

OVER is overshoot

Freq Abs, Phase P2R Diff, Phase P2P Diff

`<y> = 1 to 8`; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

`<string>`

## Examples

`SENSe:PULSE:MEASview1:ANALYZE:MEASUREMENT:TIME:TYPE FDL` sets the Measurement(s) parameter in the Define tab of the Settings control panel to Freq-domain Linearity.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:ALENgth**

Sets or queries the absolute amount of time included in the Overshoot measurements.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:ALENgth <value>
[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:ALENgth?
```

### Arguments

`<value>::<Nrf>` specifies the absolute time in seconds for the measurement time.

Range: 1 ns to 100 seconds

`<y> = 1 to 8`; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

`<Nrf>`

## Examples

`[SENSe]:PULSE:MEASview2:ANALYZE:OVERSHOOT:MEASUREMENT:TIME:ALENGTH 1E-6` sets the measurement time to 1  $\mu$ s.

## **[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:METhod**

Sets or queries the measurement time method for the Overshoot measurements.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:METHOD { ABSolute | RELative }
```

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:METHOD?
```

## Arguments

ABSolute specifies the absolute time in seconds based on rising edge of the pulse.

RELative specifies the relative time in percent based on the pulse width and the center of the pulse.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSE:PULSE:MEASview2:ANALYZE:OVERSHOOT:MEASUREMENT:TIME:METHOD REL sets Overshoot measurements to use the relative time method.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLENgth

Sets or queries the relative amount of time included in the Overshoot measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLENgth <value>
```

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLENgth?
```

## Arguments

<value> : : <NRf> specifies the relative time in percent for the measurement time based on the pulse width and centered of the pulse.

Range: 0.01% to 100%.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSE:PULSE:MEASview2:ANALYZE:OVERSHOOT:MEASUREMENT:TIME:RLENGTH 70` sets the measurement time to 70% of the pulse width.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLEVel

Sets or queries the measurement reference level for the Overshoot measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

`[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLEVel <value>`

`[SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:RLEVel?`

## Arguments

<value>::<NRf> specifies the reference level in percentage as the location on the voltage straight-line approximation of the pulse rising edge.

Range: 50% to 100%.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSE:PULSE:MEASview2:ANALYZE:OVERSHOOT:MEASUREMENT:TIME:RLEVEL 60` sets the reference level to 60% of the rising edge.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:START

Sets or queries the start time for the Overshoot measurements. The start time is the amount of time wait after the rising-edge reference before starting to measure.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:START <value>  
[SENSE]:PULSE[:MEASview<y>]:ANALyze:OVERshoot:MEASurement:TIME:START?
```

## Arguments

<value>::<NRF> specifies the start time in seconds from the reference level of the rising edge of the pulse.

Range: -100 ms to 100 ms.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRF>

## Examples

SENSE:PULSE:MEASview2:ANALYZE:OVERSHOOT:MEASUREMENT:TIME:START 1E-6 sets the start time to 1  $\mu$ s to 60% of the rising edge.

## [SENSE]:PULSE[:MEASview<y>]:ANALyze:PMLocation

Sets or queries the phase measurement location (the position along the pulse tops where the phase is measured) in the pulse-pulse phase measurement.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSE]:PULSE[:MEASview<y>]:ANALyze:PMLocation <value>  
[SENSE]:PULSE[:MEASview<y>]:ANALyze:PMLocation?
```

## Arguments

<value>::<NRF> specifies the pulse-pulse phase measurement location.

Range: 5 ns to 100 ms.



<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:ANALyze:PMLOCation 1.5ms sets the phase measurement location to 1.5 ms.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:POINT:LOCation

Sets or queries the point location method in the pulsed RF measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:POINT:LOCation { AVERage | INDePendent }
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:POINT:LOCation
```

## Arguments

AVERage uses the average amplitude calculated for the pulse-on as the 100% reference to measure the rise and fall times.

INDePendent uses the amplitudes at the beginning and end of the pulse-on as the 100% references to measure the rise and fall times, respectively.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

[SENSe]:PULSe:MEASview2:ANALyze:POINT:LOCation AVERage uses pulse average amplitude to locate points.

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:RFALI

Sets or queries the threshold levels to measure the rise/fall time.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RFALL { WIDE | NARRow }
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RFALL?
```

## Arguments

WIDE selects 10 - 90% to measure the rise/fall time.

NARRow selects 20 - 80% to measure the rise/fall time.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

```
SENSe:PULSe:MEASview2:ANALyze:RFALL WIDE selects 10 - 90% to measure the rise/fall time.
```

## [SENSe]:PULSe[:MEASview<y>]:ANALyze:RIPPlE

Sets or queries the ripple portion of the pulse top (how much of the beginning and end of the pulse top to exclude from the ripple calculation).

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RIPPlE <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:ANALyze:RIPPlE?
```

## Arguments

<value>::<NRf> specifies the ripple portion of the pulse top.

Range: 10 to 100% in 1% steps.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe:MEASview2:ANALyze:RIPPlE 30` sets the ripple portion to 30% of the pulse top.

## [SENSe]:PULSe:CARRier:FREQuency? (Query Only)

Queries the carrier frequency.

### Conditions

Measurement views: Pulse table

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe:CARRier:FREQuency?
```

### Related Commands

```
[SENSe]:PULSe:CARRier:OFFSet
```

### Arguments

None

### Returns

<NRf> indicates the actual carrier frequency in Hz.

## Examples

`SENSe:PULSe:CARRIER:FREQUENCY?` might return `2.7000001690E+9`, indicating that the pulse carrier frequency is 2.700000169 GHz.

## [SENSe]:PULSe[:MEASview<y>]:CARRier:OFFSet

Sets or queries the carrier frequency offset.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:CARRier:OFFSet <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:CARRier:OFFSet?
```

## Related Commands

[SENSe]:PULSe[:MEASview<y>]:CARRier:OFFSet

## Arguments

<value>::<NRf> specifies the carrier frequency offset.

Range: -50 kHz to +50 kHz.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:CARRier:OFFSet 2.5kHz sets the offset frequency to 2.5 kHz to the carrier.

## [SENSe]:PULSe[:MEASview<y>]:CARRier:SEARCh

Sets or queries how to detect the carrier in the pulsed RF measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

[SENSe]:PULSe[:MEASview<y>]:CARRier:SEARCh { AUTO | MANual }

[SENSe]:PULSe[:MEASview<y>]:CARRier:SEARCh?

## Arguments

AUTO specifies that the carrier is detected automatically.

MANual specifies that the carrier frequency offset is set manually, using the [SENSe]:PULSe:CARRier:OFFSet command.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:CARRier:SEARCh AUTO specifies that the carrier is detected automatically.

## **[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:BIN:COUNT**

Sets the Bins value in Histogram settings table in the Cumulative Histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:BIN:COUNT <arg1>
```

### **Arguments**

<arg1> <arg1> 100 sets the bin's value in the histogram settings table to 100 which is under Histogram Settings table in Pulse Cumulative Histogram control panel.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf1>

### **Examples**

SENSe:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:BIN:COUNT 100 sets the bin's value in the histogram table to 100.

## **[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:CLEar:RESults (Query Only)**

Clears all scalar results in the Cumulative Histogram display.

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:CLEar:RESults
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

`SENSe:PULSe:CUMULATIVE:HISTOGRAM:MEASview1:CLEAR:RESULTS` clears the result in Cumulative Histogram display.

## **[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:MODE**

Sets the Histogram settings mode to Auto or Manual in the Cumulative Histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Sense commands

## Syntax

`SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:MODE <arg1>`

## Arguments

`<arg1>` `<arg1>` AUTO or MAN sets the histogram settings mode to Auto or Manual which is under Histogram Settings table in Pulse Cumulative Histogram control panel.

## Returns

`<string>`

## Examples

`SENSe:PULSe:CUMULATIVE:HISTOGRAM:MEASview1:MODE AUTO` sets the histogram settings mode to Auto

## **[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:ORDinate (Query Only)**

Sets or queries the Histogram readout in the Cumulative Histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Sense commands

## Syntax

`SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:ORDinate <arg1>`

`SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:ORDinate?`

## Arguments

`<arg1>` `<arg1>` COUN or PERC sets the histogram readout to Count or % which is under Prefs tab in Pulse Cumulative Histogram control panel.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<string>

## Examples

SENSe:PULSe:CUMULATIVE:HISTOGRAM:MEASview1:ORDINATE ORDINATE COUN sets the histogram readout to Count.

## [SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLier:ORDinate

Sets the outlier readout in the Cumulative Histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Sense commands

## Syntax

SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:OUTLIER:ORDINATE <arg1>

## Arguments

<arg1> <arg1> COUN or PERC sets the outlier readout to Count or % which is under Prefs tab in Pulse cumulative histogram control panel.

## Returns

<string>

## Examples

SENSe:PULSe:CUMULATIVE:HISTOGRAM:OUTLIER:MEASview1:ORDINATE COUN sets the outlier readout to Count.

## [SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:SETtings:APPLY (No Query Form)

Applies the new settings to the Cumulative Histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:SETtings:APPLY
```

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Examples

SENSe:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:SETTINGS:APPLY applies new settings to the Cumulative Histogram display under Histogram Settings table in Pulse Cumulative Histogram control panel.

## [SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:XLIMit

Sets the Xlimit in Histogram settings table in the Cumulative Histogram display.

## Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe:CUMulative:HISTogram[:MEASview<y>]:XLIMit <arg1>,<arg2>,<arg3>,<arg4>
```

## Arguments arg1

arg1 can have one of the following result type:

```
{ AVERage | PPOWer | ATX | WIDTH | RISE | FALL | RINTerval | RRATe | DUTPct  
| DUTRatio | RIPple | RIPDb | DROop | DRODb | OVERshoot | OVEDb | PPPHase |  
PPFRequency | RMSFreqerror | MFReqerror | RMSPherror | MPHerror | FRDeviation |  
FDELta | PHDeviation | IRAMplitude | TIME | IRTIME | PPPD | PPFd | FABS }
```

## arg2

0 unchecks the box 1 checks the box.

Where

<arg1> is the Pulse result type.

<arg2> is the active flag which enables the limit state ON.

<arg3> is the minimum value.

<arg4> is the maximum value.

## Examples

SENSe:PULSE:CUMULATIVE:HISTOGRAM:MEASview1:XLIMIT AVG,0,10,65 sets the Pulse result type, Active flag, Minimum value and Maximum value in the histogram table.



## **SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:SELEcted:SOURce**

Returns the Source number of specified Cumulative Histogram trace

### **Conditions**

Measurement views: Cumulative Histogram with the specified Measurement View Number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### **Group**

Sense commands

### **Syntax**

```
SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:SELEcted:SOURce {"Source 1"|"Source 2"|"Source 3"|"Source 4"}
```

```
SENSe:PULSe:CUMulative:HISTogram[:MEASview<y>]:SELEcted:SOURce? (QUERY)
```

### **Arguments**

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### **Returns**

<NRf> Source Number of the specified Spectrogram trace

### **Examples**

SENSe:PULSe:CUMulative:HISTogram:MEASview1:SELEcted:SOURce? might return Source 2 for the trace of Cumulative Histogram measurement view 1

## **[SENSe]:PULSe:CUMulative:STATistics[:MEASview<y>]:CLEar:RESults**

Clears all scalar results in the Cumulative Statistics display.

### **Conditions**

Measurement views: Pulse Cumulative Statistics

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe:CUMulative:STATistics[:MEASview<y>]:CLEar:RESults
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

`SENSe:PULSe:CUMulative:STATistics:MEASview1:CLEar:RESults` clears the result in Cumulative Statistics display.

## **[SENSe]:PULSe:CUMulative:STATistics[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the selected source for Pulse Cumulative statistics measurement.

### Conditions

Measurement views: Pulse Cumulative Statistics

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe:CUMulative:STATistics[:MEASview<y>]:SELEcted:SOURce {"Source 1"  
|"Source 2"|"Source 3"|"Source 4"}
```

```
[SENSe]:PULSe:CUMulative:STATistics[:MEASview<y>]:SELEcted:SOURce?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

```
{"Source 1"|"Source 2"|"Source 3"|"Source 4"}
```

### Examples

`SENSe:PULSe:CUMulative:STATistics:MEASview1:SELEcted:SOURce?` Might return source 3, which indicates the channel of measurement source for RF data type.

## **[SENSe]:PULSe[:MEASview<y>]:DETEct:MEASurement**

Determines whether or not to set the maximum number of pulses to measure within the analysis time.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:MEASurement { OFF | ON | 0 | 1 }
[SENSe]:PULSe[:MEASview<y>]:DETECT:MEASurement?
```

## Arguments

OFF or 0 measures all pulses (max. 1000) in the analysis time.

ON or 1 specifies that the maximum number of pulses is set manually, using the `[SENSe]:PULSe:DETECT:NUMBER` command.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe:MEASview2:DETECT:MEASurement ON` specifies that the maximum number of pulses is set manually.

## [SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER

Sets or queries the maximum number of pulses to measure within the analysis time when `[SENSe]:PULSe:DETECT:MEASurement` is On.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER <value>
[SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER?
```

## Arguments

<value>: : <NRf> specifies the maximum number of pulses to measure within the analysis time. Range: 1 to 1000.

If the analysis time contains fewer pulses than this number, all of these are measured.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe:MEASview2:DETECT:NUMBER 850` sets the maximum number of pulses to 850.

## **[SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER:THReshold**

Sets or queries the power threshold to detect pulses.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER:THReshold <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:NUMBER:THReshold?
```

### Arguments

`<value>::<NRf>` specifies the power threshold to detect pulses.

Range: -70 to 0 dB.

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

`<NRf>`

## Examples

`SENSe:PULSe:MEASview2:DETECT:NUMBER:THReshold -20` sets the power threshold to -20 dB.

## **[SENSe]:PULSe[:MEASview<y>]:DETECT:TIME[:THReshold]**

Sets or queries the minimum off-time between pulses.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:TIME[:THReshold] <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:DETECT:TIME[:THRESHOLD]?
```

## Arguments

<value> : :<NRf> specifies the minimum off-time between pulses.

Range: 1 ns to 100 ms.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:DETECT:TIME:THRESHOLD 1.5ms sets the time threshold to 1.5 ms.

## [SENSe]:PULSe[:MEASview<y>]:FILTER:{BANDwidth|BWIDth}

```
[SENSe]:PULSe[:MEASview<y>]:FILTER:{BANDwidth | BWIDth}
```

Sets or queries the filter or acquisition bandwidth when [SENSe]:PULSe:FILTer:MEASurement is set to GAUSSian or NONE (No filter), respectively.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:FILTer:{BANDwidth|BWIDth} <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:FILTer:{BANDwidth|BWIDth}?
```

## Related Commands

```
[SENSe]:PULSe:FILTer:MEASurement
```

## Arguments

<value> : :<NRf> specifies the filter/acquisition bandwidth depending the [SENSe]:PULSe:FILTer:MEASurement setting. The table below shows the setting range. You can enter any value, but it is rounded up to the next valid number.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

[SENSe]:PULSe:FILTer:MEASurement	Range
GAUSSian	100 Hz to 20 MHz (Standard) / 100 Hz to 55 MHz (Option 110) in 1-2-3-5 sequence.

Table continued...

[SENSe]:PULSe:FILTer:MEASurement	Range
NONE	152, 305, 610, 1.22 k, 2.44 k, 4.88 k, 9.76 k, 19.3 k, 39 k, 78 k, 156 k, 312 k, 625 k, 1.25 M, 2.5 M, 5 M, 10 M, 20 M, 40 MHz, and optionally 60 M and 110 MHz (Option 110).

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:FILTer:BANDwidth 10MHz sets the filter bandwidth to 10 MHz.

## [SENSe]:PULSe[:MEASview<y>]:STATistics:SELEcted:SOURce

Sets or queries selected source for pulse statistics measurement.

## Conditions

Measurement views: Pulse Statistics

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:SELEcted:SOURce {"Source 1"|"Source 2"|"Source 3"|"Source 4"}
```

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:SELEcted:SOURce?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:STATistics:SELEcted:SOURce? queries the source of the Pulse Statistics Measurement View 2.

## [SENSe]:PULSe[:MEASview<y>]:FILTer:MEASurement

Sets or queries the measurement filter in the pulsed RF measurements.

## Conditions

Measurement views: Pulse statistics, Pulse table, Pulse trace

## Group

Sense commands

## Syntax

```
[SENSE]:PULSE[:MEASview<y>]:FILTER:MEASUREMENT { GAUSSian | NONE | MAXRtbw }
```

```
[SENSe]:PULSE[:MEASview<y>]:FILTER:MEASUREMENT?
```

## Related Commands

```
[SENSe]:PULSE[:MEASview<y>]:FILTER { BANDwidth|BWIDth }
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

GAUSSian uses the Gaussian filter in the pulsed RF measurements. Use the [SENSE]:PULSE[:MEASview<y>]:FILTER:{BANDwidth|BWIDth} command to set the filter bandwidth.

NONE uses no filter. Use the [SENSe]:PULSE[:MEASview<y>]:FILTER:{BANDwidth|BWIDth} command to set the acquisition bandwidth.

MAXRtbw uses no filter. The acquisition bandwidth is fixed to the maximum real-time bandwidth: 40 MHz (Standard) or 110 MHz (Option 110).

## Examples

```
SENSe:PULSE:MEASview1:FILTER:MEASUREMENT GAUSSian uses the Gaussian filter in the pulsed RF measurements.
```

## [SENSe]:PULSE[:MEASview<y>]:FREference:AUTO

Determines whether to estimate the pulse frequency reference automatically or manually in the pulsed RF measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSE[:MEASview<y>]:FREference:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:PULSE[:MEASview<y>]:FREference:AUTO?
```

## Arguments

OFF or 0 estimates the frequency reference manually. Use the `[SENSe]:PULSe:FREFerence:OFFSet` command to set the frequency offset.

ON or 1 estimates the frequency reference automatically.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe[:MEASview2]:FREFerence:AUTO ON` specifies that the maximum number of pulses is set manually.

## [SENSe]:PULSe[:MEASview<y>]:FREFerence:CSLope

Sets or queries the Chirp Slope.

Chirp Slope is used when the Modulation Type is set to Linear Chirp. When set, this also changes the Chirp Slope to manual mode. When chirp slope is in auto mode, the query returns the estimated chirp slope.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:FREFerence:CSLope <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:FREFerence:CSLope?
```

## Arguments

<value>: = NRf specifies the chirp slope in Hz/μs.

Range: -100 GHz/μs to 100 GHz/μs.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe:MEASview2:FREFerence:CSLope 1E6` sets the chirp slope to 1 MHz/μs.



## [SENSe]:PULSe[:MEASview<y>]:FREFeRence:CSLoPe:AUTO

Sets or queries enabling automatic Chirp Slope estimating in the pulsed RF measurements.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:FREFeRence:CSLoPe:AUTO { OFF | ON | 0 | 1 }
[SENSe]:PULSe[:MEASview<y>]:FREFeRence:CSLoPe:AUTO?
```

### Arguments

OFF or 0 estimates the chirp slope manually. Use the [SENSe]:PULSe[:MEASview<y>]:FREFeRence:CSLoPe command to set the chirp slope.

ON or 1 estimates the chirp slope automatically.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

### Examples

SENSe:PULSe:MEASview2:FREFeRence:CSLoPe:AUTO ON sets the chirp slope to auto mode.

## [SENSe]:PULSe[:MEASview<y>]:FREFeRence:OFFSet

Sets or queries the frequency reference offset. This command is valid when

[SENSe]:PULSe[:MEASview<y>]:FREFeRence:AUTO is set to OFF.

### Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe[:MEASview<y>]:FREFeRence:OFFSet <value>
[SENSe]:PULSe[:MEASview<y>]:FREFeRence:OFFSet?
```

## Arguments

<value> : : <NRf> specifies the frequency reference offset.

Range: -50 kHz to +50 kHz.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:FREference:OFFSet 2.5kHz sets the frequency offset to 2.5 kHz to the reference.

## [SENSe]:PULSe[:MEASview<y>]:MODulation:TYPE

Sets or queries the modulation type in the pulsed RF measurements. This command is equivalent to the [SENSe]:REANalyze:NEXT command.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:MODulation:TYPE { CWConst | CWCHange | LCHirp | OTHer}
```

```
[SENSe]:PULSe[:MEASview<y>]:MODulation:TYPE?
```

## Arguments

CWConst selects the CW (continuous wave) with constant phase.

CWCHange selects the CW (continuous wave) with changing phase.

LCHirp selects the linear chirp.

OTHer selects other.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:MODulation:TYPE CWCHang selects the CW with changing phase as the modulation type.

## **[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:FILTer:SHAPE**

Sets or queries the type of Filter shape from Analysis tab of Frequency vs Time from the Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:FILTer:SHAPE <arg1>
```

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:FILTer:SHAPE?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASView keyword is optional by default y is 1.

<arg1> is KAIS or UNIF or BH4B or FLAT or HAMM

Where

UNIF is Uniform (none).

BH4B is Blackman-Harris 4B.

FLAT is Flat - Top (CW ampl).

KAIS is Kaiser (RBW).

HAMM is Hamming.

### **Returns**

<string> KAIS or UNIF or BH4B or FLAT or HAMM

### **Examples**

SENSe:PULSe:OGRAM:MEASview1:ANALysis:FVTime:FILTer:SHAPEKAIS sets the Analysis Filter shape to Kaiser(RBW).

## **[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:SPECTrum:ANALysis:REGion**

Sets or queries the Spectrum Analysis region from Analysis tab of Frequency vs Time in the Pulse-Ogram display.

### **Conditions**

Measurement views: Pulse-Ogram

### **Group**

Sense commands

## Syntax

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:SPECTrum:ANALysis:REGion <arg1>  
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:SPECTrum:ANALysis:REGion?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<arg1> is ATIM or POT or IND

Where

ATIM is Use Analysis Time settings.

POT is Pulse ON Time.

IND is Independent.

## Returns

<string> ATIM or POT or IND

## Examples

SENSe:PULSe:OGRAM:MEASview1:ANALysis:FVTime:SPECTrum:ANALysis:REGionATIM sets the Spectrum Analysis region to Use Analysis Time settings option.

## [SENSe]:PULSe:OGRAM[:MEASview<y>]:SELEcted:SOURce

Returns the Source number of specified pulso-gram trace.

## Conditions

Measurement views: Pulse-Ogram

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:SELEcted:SOURce {"Source 1"|"Source 2"|"Source  
3"|"Source 4"}  
[SENSe]:PULSe:OGRAM[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

{"Source 1"|"Source 2"|"Source 3"|"Source 4"}

## Examples

`SENSe:PULSe:OGRAM:MEASview1:SELEcted:SOURce?` queries the source of the PulseOgram measurement view 1.

## **[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:ONTime:LENgth**

Sets or queries the OnTime value from Analysis tab of Frequency vs Time in the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:ONTime:LENgth <value>
```

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:ONTime:LENgth?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf>

### Returns

<NRf> specifies the ON time in seconds.

### Examples

`SENSe:PULSe:OGRAM:MEASview<y>:ANALysis:FVTime:ONTime:LENgth?` might return 10E-6s.

## **[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:STARt**

Sets or queries the independent spectrum start time from Analysis tab of Frequency vs Time in the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:STARt <value>
```

```
[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:STARt?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf>

## Returns

<NRf> specifies the independent spectrum start time in seconds.

## Examples

SENSe:PULSe:OGRAM:MEASview1:ANALysis:FVTime:INDEpendent:STARt? might return 10E-6s.

## [SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:LENgth

Sets or queries the independent spectrum length from Analysis tab of Frequency vs Time in the Pulse-Ogram display.

## Conditions

Measurement views: Pulse-Ogram

## Group

Sense commands

## Syntax

[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:LENgth <value>

[SENSe]:PULSe:OGRAM[:MEASview<y>]:ANALysis:FVTime:INDEpendent:LENgth?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf>

## Returns

<NRf> specifies the independent spectrum length in seconds.

## Examples

SENSe:PULSe:OGRAM:MEASview1:ANALysis:FVTime:INDEpendent:LENgth? might return 10E-6s.

## [SENSe]:PULSe[:MEASview<y>]:SIGNal:TYPE

Sets or queries the signal type in the pulsed RF measurements.

## Conditions

Measurement views: Pulse Statistics with specified Measurement view number, Pulse table, Pulse trace with specified Measurement view number

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:SIGNal:TYPE { CWConst | CWCHange | LCHirp | OTHer}
[SENSe]:PULSe[:MEASview<y>]:SIGNal:TYPE?
```

## Arguments

`CWConst` selects CW (continuous wave) with constant phase.

`CWCHange` selects CW (continuous wave) with changing phase.

`LCHirp` selects linear chirp.

`OTHer` selects other.

`<y>` = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

`SENSe:PULSe:MEASview2:SIGNal:TYPE CWCHange` selects the CW with changing phase as the signal type.

## [SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:ORDinate

Sets or queries the Histogram Y unit. This command is valid when `DISPlay:PULSe:STATistics:PLOT` is set to Histogram.

## Conditions

Measurement views: Pulse statistics

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:ORDinate { PERCent | COUNT }
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:ORDinate?
```

## Arguments

`PERCent` sets the Histogram plot Y axis to the percentage of points.

`COUNT` sets the Histogram plot Y axis to the number of points in the bin.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:STATistics:HISTogram:ORDinate PERCENT sets the histogram Y axis to percent.

## [SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs

Sets or queries the number of bins in the Histogram plot.

## Conditions

Measurement views: Pulse statistics

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs <value>
```

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs?
```

## Arguments

<value> ::= <NRf> specifies the number of histogram bins.

Range: 5 to 10000

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:STATistics:HISTogram:POINTs 10 sets the number of histogram bins to 10.

## [SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs:AUTO

Sets or queries enabling automatic Histogram plot bins count.

## Conditions

Measurement views: Pulse statistics



## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs:AUTO { OFF | ON | 0 | 1 }
[SENSe]:PULSe[:MEASview<y>]:STATistics:HISTogram:POINTs:AUTO?
```

## Arguments

OFF or 0 sets the number of bins manually by the [SENSe]:PULSe:STATistics:HISTogram:POINTs command.

ON or 1 sets the number of bins to the number of pulses/10.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

SENSe:PULSe:MEASview2:STATistics:HISTogram:POINTs:AUTO 1 sets the number of histogram bins based on number of pulses.

## [SENSe]:PULSe[:MEASview<y>]:STATistics:IN:LINear:SCALE

Sets or queries the "Calculate statistics in linear scale" selection from the display.

## Conditions

Measurement views: Pulse Cumulative Statistics, Pulse Statistics

## Group

Sense commands

## Syntax

```
[SENSe]:PULSe[:MEASview<y>]:STATistics:IN:LINear:SCALE { 0 | OFF | 1 | ON}
[SENSe]:PULSe[:MEASview<y>]:STATistics:IN:LINear:SCALE?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

0 or OFF unchecks the box and disables the calculate function

1 or ON checks the box and enables the calculate function

## Returns

<Boolean>, 0 or 1

## Examples

`SENSe:PULSe:MEASview1:STATistics:IN:LINear:SCALE 1` enables the Calculate statistics in linear scale (for results in log).

## **[SENSe]:PULSe:TRACe[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source of the selected Measurement view in pulse trace.

### Conditions

Measurement views: Pulse trace with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

```
[SENSe]:PULSe:TRACe[:MEASview<y>]:SELEcted:SOURce {"Source 1"|"Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:PULSe:TRACe[:MEASview<y>]:SELEcted:SOURce?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

## Examples

`SENSe:PULSE:TRACE:MEASVIEW1:SELECTED:SOURCE?` queries the source of the pulse trace measurement view 1.

## **[SENSe]:PULSe:RESult[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source of the selected Measurement view in pulse Result.

### Conditions

Measurement views: Pulse Result with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

## Syntax

```
[SENSe]:PULSe:RESult[:MEASview<y>]:SELEcted:SOURce
{"Source1"|"Source2"|"Source3"|"Source4"}

[SENSe]:PULSe:RESult[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

<y>=1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1

## Examples

```
[SENSe]:PULSe:RESult[:MEASview1]:SELEcted:SOURce? Queries the Source of the spectrum Measurement View 1.
```

## [SENSe]:REANalyze (No Query Form)

Sets all measurements to reanalyze the last acquisition.



**Note:** *hg* is an overlapped command, which does not finish executing before the next command starts executing. Use the *\*OPC(?)* and *\*WAI* commands to synchronize all pending operations to the execution of this command.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze
```

## Related Commands

\*OPC, \*WAI

## Arguments

None

## Examples

SENSe:REANALYZE reanalyzes all measurements using the current acquisition.

## [SENSe]:REANalyze:ACQuisition:SETTings? (Query Only)

Queries the settings of all acquisitions.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:REANalyze:ACQuisition:SETTings?

## [SENSe]:REANalyze:ALL:LOOP (No Query Form)

Replay all selected records/frames in a continuous loop.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:REANalyze:ALL:LOOP

## [SENSe]:REANalyze:CURRent (No Query Form)

Replay the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:REANalyze:CURRent

## [SENSe]:REANalyze:CURRent:ACQuisition? (Query Only)

Queries the Acquisition record identifier (ID) of the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

---

## Syntax

```
[SENSe]:REANalyze:CURRent:ACQuisition?
```

## [SENSe]:REANalyze:CURRent:FRAMe? (Query Only)

Queries the Frame identifier (ID) of the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:CURRent:FRAMe?
```

## [SENSe]:REANalyze:CURRent (No Query Form)

Replay the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:CURRent
```

## [SENSe]:REANalyze:CURRent:TIMestamp? (Query Only)

Queries the time stamp of the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:CURRent:TIMestamp?
```

## **[SENSe]:REANalyze:DELeTe:DATA (No Query Form)**

Deletes all acquisition data.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:DELeTe:DATA
```

## **[SENSe]:REANalyze:FIRSt (No Query Form)**

Replays (reanalyzes) the first item in the currently selected acquisition (or frame).

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:FIRSt
```

## **[SENSe]:REANalyze:LAST (No Query Form)**

Replays (reanalyzes) the last item in the currently selected acquisition (or frame).

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:LAST
```

## **[SENSe]:REANalyze:NEXT (No Query Form)**

Replays (reanalyzes) the next item in the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:NEXT
```

## [SENSe]:REANalyze:PAUSE

Sets or queries the reanalysis paused state.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:PAUSE {OFF | ON | 0 | 1 }
```

```
[SENSe]:REANalyze:PAUSE?
```

## Arguments

ON or 1 pauses (blocks) any acquisition until current reanalysis is completed.

OFF or 0 disables the reanalysis pause mode.

## [SENSe]:REANalyze:PREVIOUS (No Query Form)

Replays (reanalyzes) the previous item in the currently selected acquisition (or frame).

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:PREVIOUS
```

## [SENSe]:REANalyze:SElect:ACQuisition:DATA:TYPE

Sets or queries the acquisition data type.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:DATA:TYPE {SPEctra | IQ }
```

```
[SENSe]:REANalyze:SElect:ACQuisition:DATA:TYPE?
```

## Arguments

SPEctra sets the acquisition data type as DPX spectra data.

IQ sets the acquisition data type a IQ data.

## Examples

SENSe:REANALYZE:SELECT:ACQUISITION:DATA:TYPE IQ sets the acquisition data type to IQ data.

## [SENSe]:REANalyze:SElect:ACQuisition:FIRSt? (Query Only)

Queries the first selectable acquisition Start or Stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt:TIMestamp?
```

## [SENSe]:REANalyze:SElect:ACQuisition:FIRSt:TIMestamp? (Query Only)

Queries the time stamp of the first selectable acquisition history Start or Stop value in the currently selected acquisition set.

## Conditions

Measurement views: All

## Group

Sense commands



## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt:TIMestamp?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:ACQuisition:FIRSt?
```

## [SENSe]:REANalyze:SElect:ACQuisition:LAST? (Query Only)

Queries the last selectable acquisition Start or Stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:LAST?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:ACQuisition:LAST:TIMestamp?
```

## [SENSe]:REANalyze:SElect:ACQuisition:LAST:TIMestamp? (Query Only)

Queries the time stamp of the last selectable acquisition Start or Stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:LAST:TIMestamp?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:ACQuisition:LAST?
```

## [SENSe]:REANalyze:SElect:ACQuisition:START

Sets or queries the Acquisition Start value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:START <value>
```

```
[SENSe]:REANalyze:SElect:ACQuisition:START?
```

## Arguments

<value>: : <NR1> is an integer in the range from one to the number of acquisitions in the acquisition history.

## [SENSe]:REANalyze:SElect:ACQuisition:START:TIMestamp? (Query Only)

Queries the time stamp of the current Acquisition Start selection. It is the time stamp of the first frame of the selected acquisition, if FastFrame is enabled.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:START:TIMestamp?
```

## [SENSe]:REANalyze:SElect:ACQuisition:STOP

Sets or queries the Acquisition Stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:ACQuisition:STOP <value>
```

```
[SENSe]:REANalyze:SElect:ACQuisition:STOP?
```

## Arguments

<value>: : <NR1> is an integer in the range from one to the number of acquisitions in the acquisition history.

---

## **[SENSe]:REANalyze:SElect:ACQuisition:STOP:TIMestamp? (Query Only)**

Queries the time stamp of the current Acquisition Stop selection. This is the time stamp of the last frame of the selected acquisition, if FastFrame is enabled.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:ACQuisition:STOP:TIMestamp?
```

## **[SENSe]:REANalyze:SElect:ALL (No Query Form)**

Selects all acquisitions and frames in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:ALL
```

## **[SENSe]:REANalyze:SElect:FRAMe:FIRSt? (Query Only)**

Queries the first available Frame Start or Stop value available in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:FRAMe:FIRSt?
```

### **Related Commands**

```
[SENSe]:REANalyze:SElect:FRAMe:FIRSt:TIMestamp?
```

## **[SENSe]:REANalyze:SElect:FRAMe:FIRSt:TIMestamp? (Query Only)**

Queries the time stamp of the first available frame of the first available Acquisition Start or Stop value available in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:FRAMe:FIRSt:TIMestamp?
```

### **Related Commands**

```
[SENSe]:REANalyze:SElect:FRAMe:FIRSt?
```

## **[SENSe]:REANalyze:SElect:FRAMe:LAST? (Query Only)**

Queries the last available Frame Start or Stop value available in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:FRAMe:LAST?
```

### **Related Commands**

```
[SENSe]:REANalyze:SElect:FRAMe:LAST:TIMestamp?
```

## **[SENSe]:REANalyze:SElect:FRAMe:LAST:TIMestamp? (Query Only)**

Queries the time stamp of the last available frame of the last available Acquisition Start or Stop value available in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:FRAMe:LAST:TIMestamp?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:FRAMe:LAST?
```

## [SENSe]:REANalyze:SElect:FRAMe:START

Sets or queries the Frame Start value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:FRAMe:START <value>
```

```
[SENSe]:REANalyze:SElect:FRAMe:START?
```

## Arguments

<value>::<NR1> is an integer in the range from one to the number of fast frames.

## [SENSe]:REANalyze:SElect:FRAMe:START:TIMestamp? (Query Only)

Queries the time stamp of the current Frame Start selection. This is the time stamp of the Frame Start frame of the acquisition start acquisition, if FastFrame is enabled.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:FRAMe:START:TIMestamp?
```

## [SENSe]:REANalyze:SElect:FRAMe:STOP

Sets or queries the frame stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:FRAMe:STOP <value>
```

```
[SENSe]:REANalyze:SElect:FRAMe:STOP?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:FRAMe:STOP:TIMEstamp?
```

## Arguments

<value>: : <NR1> is an integer in the range from one to the number of fast frames.

## [SENSe]:REANalyze:SElect:FRAMe:STOP:TIMEstamp? (Query Only)

Queries the time stamp of the current frame stop selection. It is the time stamp of the frame Stop frame of the Acquisition Stop acquisition, if FastFrame is enabled.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:FRAMe:STOP:TIMEstamp?
```

## Related Commands

```
[SENSe]:REANalyze:SElect:FRAMe:STOP
```

## [SENSe]:REANalyze:SElect:SPECTra:ACQquisition:ALL (No Query Form)

Selects all spectra acquisitions and frames in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQquisition:ALL
```

## **[SENSe]:REANalyze:SElect:SPECTra:ACQquisition:CURRent:ACQquisition? (Query Only)**

Queries the acquisition record identifier (ID) of the currently selected acquisition.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:SPECTra:ACQquisition:CURRent: ACQquisition?
```

### **Returns**

<ID Number> ::= <NR1> the record identifier number of the current acquisition.

### **Examples**

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:CURRENT:ACQUISITION? might return 848 indicating that the current acquisition record number is 848.

## **[SENSe]:REANalyze:SElect:SPECTra:ACQquisition:CURRent:TIMestamp? (Query Only)**

Queries the time stamp of the currently selected acquisition.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:SPECTra:ACQquisition:CURRent: TIMestamp?
```

### **Returns**

<time stamp> ::= <string> The current time stamp value.

### **Examples**

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:CURRENT:TIMESTAMP? might return "10/10/2011 10:57:55.93" indicating that the current time stamp recorded on October 10, 2011 was 10 hours, 57 minutes, and 55.93 seconds.

## **[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:FIRSt? (Query Only)**

Queries the first selectable acquisition start or stop value in the currently selected set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:FIRSt?
```

### **Returns**

<ID Number> ::= <NR1> the record identifier number of the current acquisition.

### **Examples**

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:FIRST? might return 1 indicating that the first selectable acquisition start or stop value was 1.

## **[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:FIRSt:TIMestamp? (Query Only)**

Queries the time stamp of the first selectable acquisition history start or stop value in the currently selected acquisition set.

### **Conditions**

Measurement views: All

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:FIRSt:TIMestamp?
```

### **Returns**

<time stamp> ::= <string> The first time stamp value.

### **Examples**

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:FIRST:TIMESTAMP? might return "10/10/2011 10:57:41:58" indicating that the first time stamp recorded on October 10, 2011 was 10 hours, 57 minutes, and 41.58 seconds.

## **[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:LAST? (Query Only)**

Queries the last selectable acquisition start or stop value in the currently selected set.



## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:LAST?
```

## Returns

<ID Number> ::= <NR1> the record identifier number of the current acquisition.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:LAST? might return 1 indicating that the last selectable acquisition start or stop value was 1.

## [SENSe]:REANalyze:SElect:SPECTra:ACQuisition:LAST:TIMestamp? (Query Only)

Queries the time stamp of the last selectable acquisition start or stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:LAST:TIMestamp?
```

## Returns

<time stamp> ::= <string> The last time stamp value.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:LAST:TIMESTAMP? might return "10/10/2011 10:57:41:58" indicating that the last time stamp recorded on October 10, 2011 was 10 hours, 57 minutes, and 41.58 seconds.

## [SENSe]:REANalyze:SElect:SPECTra:ACQuisition:START

Sets or queries the acquisition start value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:START <value>
```

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:START?
```

## Arguments

<value> ::= <NR1> is an integer in the range from one to the number of acquisitions in the acquisition history.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:START? might return 1.0000000000 indicating the start value was 1.

## [SENSe]:REANalyze:SElect:SPECTra:ACQuisition:START:TIMestamp? (Query Only)

Queries the time stamp of the current acquisition start selection.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:START:TIMestamp?
```

## Returns

<time stamp> ::= <string> The start time stamp value.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:START:TIMESTAMP? might return "10/10/2011 10:57:41:58" indicating that the time stamp of the acquisition start selection recorded on October 10, 2011 was 10 hours, 57 minutes, and 41.58 seconds.

## [SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP

Sets or queries the acquisition stop value in the currently selected set.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP <value>
```

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP?
```

## Arguments

<value> ::= <NR1> is an integer in the range from one to the number of acquisitions in the acquisition history.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:STOP? might return 1.0000000000 indicating the stop value was 1.

## [SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP:TIMestamp? (Query Only)

Queries the time stamp of the current acquisition stop selection.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SElect:SPECTra:ACQuisition:STOP:TIMestamp?
```

## Returns

<time stamp> ::= <string> The stop time stamp value.

## Examples

SENSe:REANALYZE:SELECT:SPECTRA:ACQUISITION:STOP:TIMESTAMP? might return "10/10/2011 10:57:41.58" indicating that the time stamp of stop selection recorded on October 10, 2011 was 10 hours, 57 minutes, and 41.58 seconds.

## [SENSe]:REANalyze:SPEed

Sets or queries the time delay value inserted between reanalysis of frames or acquisitions.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:SPEEd <value>
```

```
[SENSe]:REANalyze:SPEEd?
```

## Arguments

<value>::<NR1> is an integer in the range of zero (for no delay) to four (for a maximum delay of about one second).

## [SENSe]:REANalyze:STOP (No Query Form)

Stops the current reanalysis. The command will block until the current reanalysis has completed and stopped.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:STOP
```

## [SENSe]:REANalyze:TIMestamp:DECimal

Sets or queries the number of values to the right of the decimal point to include in all time stamps.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:REANalyze:TIMestamp:DECimal <value>
```

```
[SENSe]:REANalyze:TIMestamp:DECimal?
```

## Arguments

<value>::<NR1> is an integer in the range of one to nine.

## SENSe:ROSCillator:EXTernal:TIME

Sets or queries the timing reference.

**For the RSA7100A (Option GPS required):** The default timing reference source upon launch of SignalVu-PC is system time. If you change the timing reference source from the default to one of the externally generated sources (GPS, PPS, IRIG-B) without a valid signal, the time will remain system time.

If you set the source to GPS, PPS, or IRIG-B after launch of SignalVu-PC (thus changing the time from the default system time), and then disconnect from that source or otherwise lose the lock to that source, the time used will be that of the previous externally generated source. This will remain so until you set a different source with a valid signal.

## Group

Sense commands

## Syntax

```
SENSe:ROSCillator:EXternal:TIME {PPS | IBAM | IBDC | GPS | SYStem}
SENSe:ROSCillator:EXternal:TIME?
```

## Related Commands

```
SENSe:ROSCillator:EXternal:TIME:STATus?
```

## Arguments

PPS is a 1PPS signal

IBAM is an IRIG-B AM signal

IBDC is an IRIG-B DC signal

GPS is a GNSS signal

SYStem is the CTRL7100A (PC) time

## Returns

One of the following: PPS, IBAM, IBDC, GPS, SYStem

## Examples

```
SENSe:ROSCILLATOR:EXTERNAL:TIME PPS sets the system time to 1PPS.
```

## SENSe:ROSCillator:EXternal:TIME:STATus? (Query Only)

Queries if the timing reference source is locked or unlocked.

## Group

Sense commands

## Syntax

```
SENSe:ROSCillator:EXternal:TIME:STATus?
```

## Related Commands

```
SENSe:ROSCillator:EXternal:TIME
```

## Returns

0 means the timing source is unlocked.

1 means the timing source is locked.

## Examples

`SENSe:ROSCILLATOR:EXTERNAL:TIME:STATUS?` 1 indicates the timing source is locked. This is indicated in the UI on the Timing Reference tab by a green checked box next to Valid Timing Source.

## [SENSe]:ROSCillator:SOURce

Sets or queries the frequency reference oscillator source.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:ROSCillator:SOURce { INTernal | EXTernal }
```

```
[SENSe]:ROSCillator:SOURce?
```

## Related Commands

`SENSe:ROSCillator:EXTernal:TIME:STATus?`

## Arguments

`INTernal` specifies that the analyzer uses the 10 MHz internal oscillator.

`EXTernal` specifies that the analyzer uses the external reference signal. Use the Ref In connector on the rear panel to input the signal.

Selecting `EXTernal` initiates an attempt to lock the internal reference oscillator to the external reference signal. If the signal is not connected or is at an invalid frequency or amplitude, an error (2028, "External frequency reference signal not valid. Using internal reference") is returned. If not able to lock to the external reference, an error (2029, "Unable to lock to external frequency reference. Using internal reference.") is returned.

## Examples

`SENSe:ROSCILLATOR:SOURCE INTernal` uses the 10 MHz internal oscillator for the frequency reference.

## [SENSe]:SEM:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW) of the reference channel.



**Note:** *In Real-time operation this sets the RBW for the reference and all offset channels.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:{BANDwidth|BWIDth}[:RESolution] <value>
```

```
[SENSe]:SEM:{BANDwidth|BWIDth}[:RESolution]?
```

## Arguments

<value>::<NRf> specifies the resolution bandwidth.

## Examples

SENSe:SEM:{BANDWIDTH|BWIDTh}[:RESOLUTION] 200kHz sets the resolution bandwidth to 200 kHz.

## [SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW) of the reference channel.



**Note:** *In Real-time operation this sets the RBW for the reference and all offset channels.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo <value>
```

```
[SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo?
```

## Arguments

<value>::<NRf> specifies the video bandwidth.

Range: Current RBW/104 (1 Hz minimum) to current RBW.

## Examples

SENSe:SEM:{BANDWIDTH|BWIDTh}:VIDEO 200kHz sets the video bandwidth to 200 kHz.

## [SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo:STATe

Sets or queries the state of the video bandwidth (VBW) of the reference channel.



**Note:** *In Real-time operation this sets the VBW state for the reference and all offset channels.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo:STATe?
```

## Arguments

OFF or 0 turns the video bandwidth filter off.

ON or 1 turns the video bandwidth filter on.

## [SENSe]:SEM:CHANnel:FILTer

Sets or queries the reference Channel Filter.



**Note:** *In Real-time operation the channel filter is always forced to None.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:CHANnel:FILTer { RRC | NONE }
```

```
[SENSe]:SEM:CHANnel:FILTer?
```

## Arguments

RRC selects the Root-Raised-Cosine filter.

NONE uses no filter.

## Examples

`SENSe:SEM:CHANNEL:FILTER RRCosine` selects Root-Raised-Cosine for the reference Channel Filter.

## [SENSe]:SEM:CHANnel:IBANdwidth

Sets or queries the reference channel integration bandwidth.



## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:CHANnel:IBANdwidth <value>
```

```
[SENSe]:SEM:CHANnel:IBANdwidth?
```

## Arguments

<value>::<Nrf> integration bandwidth

## Returns

## Examples

SENSe:SEM:CHANNEL:IBANDWIDTH? might return 18.000000E+6 indicating that the integration bandwidth is 18 MHz.

## [SENSe]:SEM:CHANnel:WIDTh

Sets or queries the reference Channel bandwidth.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:CHANnel:WIDTh <value>
```

```
[SENSe]:SEM:CHANnel:WIDTh?
```

## Arguments

<value>::<Nrf> specifies the filter bandwidth.

## Examples

SENSe:SEM:CHANNEL:WIDTH 30KHZ sets the filter bandwidth to 30 kHz.

## [SENSe]:SEM:CHIPrate

Sets or queries the Chip rate.

Use the [SENSe]:SEM:CHANnel:FILTer command to set the reference Channel Filter to RRC (Root-Raised Cosine).



**Note:** *Applies in Swept operation only.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:CHIPrate <value>
```

```
[SENSe]:SEM:CHIPrate?
```

## Arguments

<value>::<Nrf> specifies the chip rate.

## Examples

SENSe:SEM:CHIPRATE 3.84MHZ sets the Chip rate to 3.84 MHz.

## [SENSe]:SEM:CLEAr:RESults (No Query Form)

Clears the results in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:CLEAr:RESults
```

## Arguments

None

## Examples

SENSe:SEM:CLEAr:RESULTS clears the results in the measurement.

## [SENSe]:SEM:FREQuency

Sets or queries the measurement frequency.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:FREQuency <value>
```

```
[SENSe]:SEM:FREQuency?
```

## Arguments

<value>: : <Nrf> specifies the center frequency.

## Examples

```
[SENSe]:SEM:FREQUENCY 1.0GHZ sets the center frequency to 1.0 GHz.
```

## [SENSe]:SEM:FREQuency:SPAN? (Query Only)

Returns the frequency span.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:FREQuency:SPAN?
```

## Arguments

None

## Returns

<Nrf> frequency span.

## Examples

```
SENSe:SEM:FREQUENCY:SPAN? might return 100.00000E+6 indicating that the frequency span is 100 MHz.
```

## [SENSe]:SEM:MFILter

Sets or queries the shape of the filter.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:MFILter { GAUSSian | RECTangular }  
[SENSe]:SEM:MFILter?
```

## Arguments

GAUSSian sets the filter shape to Gaussian.

RECTangular sets the filter shape to Rectangular.

## Examples

SENSe:SEM:MFILTER RECTangular sets the filter shape to Rectangular.

## [SENSe]:SEM:MODE:REALtime

Sets or queries the Real-time SEM measurement mode.



**Note:** *In real-time mode SEM is measured using real-time acquisition data.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:MODE:REALtime { OFF | ON | 0 | 1 }  
[SENSe]:SEM:MODE:REALtime?
```

## Arguments

ON or 1 turns the SEM real-time measurement mode on.

OFF or 0 turns the SEM real-time measurement mode off.

## [SENSe]:SEM:MTYPE

Sets or queries the measurement type.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:MTYPE { TPOW | PSD | PEAK }
```

## Arguments

TPOW specifies the Total Power measurement.

PSD specifies the Peak Spectral Density measurement in dBm/Hz.

PEAK specifies a Peak power measurement.

## Examples

```
SENSe:SEM:MTYPE PEAK sets the measurement type to Peak.
```

## [SENSe]:SEM:OFFSet:DETection

Sets or queries the detection settings for all offsets.



**Note:** In Real-time operation the offset and reference detection settings are forced equal.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet:DETection { AVERage | PEAK }
```

```
[SENSe]:SEM:OFFSet:DETection?
```

## Arguments

AVERage indicates the Offsets setting is Average (VRMS).

PEAK indicates the Offsets setting is the positive peak.

## Examples

```
SENSe:SEM:OFFSET:DETECTION AVER sets the detection settings for all offsets to Average RMS.
```

## [SENSe]:SEM:OFFSet<x>:BANDwidth:FILTer? (Query Only)

Returns the bandwidth filter setting for specified row in the Offsets & Limits table. The filter setting is equal to the Resolution Bandwidth (RBW) value times the number of resolution bandwidths. The parameter <x> represents the row in the Offsets & Limits table.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Sense commands

### Syntax

```
[SENSe]:SEM:OFFSet<x>:BANDwidth:FILTer?
```

### Arguments

None

### Returns

<value>::<NRf> is the filter bandwidth setting.

### Examples

SENSe:SEM:OFFSet3:BANDwidth:FILTer might return 30.0E+3 indicating that the filter bandwidth for row 3 to 30 kHz.

## [SENSe]:SEM:OFFSet<x>:BANDwidth[:RESolution]

Sets or queries the resolution bandwidth for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table. See the [SENSe]:SEM:{BANDwidth|BWIDTH}[:RESolution] command.



**Note:** *In Real-time operation the RBW values for each offset and the reference are set equal to the All Channels RBW setting.*

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Sense commands

### Syntax

```
[SENSe]:SEM:OFFSet<x>:BANDwidth[:RESolution] <value>
```

```
[SENSe]:SEM:OFFSet<x>:BANDwidth[:RESolution]?
```

### Returns

<value>::<NRf> specifies the resolution bandwidth.

## Examples

`SENSe:SEM:OFFSet4:BANDwidth[:RESolution] 30kHz` sets the resolution bandwidth in row 4 to 30 kHz.

## [SENSe]:SEM:OFFSet<x>:BANDwidth:RESolution:COUNT

Sets or queries the count of Resolution Bandwidths (RBWs) for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.



**Note:** *In Real-time operation the RBW count is always forced to 1.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

`[SENSe]:SEM:OFFSet<x>:BANDwidth:RESolution:COUNT <value>`

`[SENSe]:SEM:OFFSet<x>:BANDwidth:RESolution:COUNT?`

## Arguments

<value> specifies the count number of RBWs.

## Examples

`SENSe:SEM:OFFSet1:BANDwidth:RESolution:COUNT 10` sets the count for row 1 to 10.

## [SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo

Sets or queries the video bandwidth setting for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table. See the `[SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo` command.



**Note:** *In Real-time operation the VBW values for each offset and the reference are set equal to the All Channels VBW setting.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

`[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo <value>`

`[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo?`

## Arguments

<value> : : <Nrf> specifies the video bandwidth setting.

## Examples

SENSe:SEM:OFFSet3:BANDwidth:VIDeo 300 sets the video bandwidth for row 3 to 300 Hz.

## [SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo:STATe

Sets or queries the state of the video bandwidth for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table. See the [SENSe]:SEM:{BANDwidth|BWIDth}:VIDeo:STATe command.



**Note:** *In Real-time operation the VBW state values for each offset and the reference are set equal to the All Channels VBW state setting.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:SEM:OFFSet<x>:BANDwidth:VIDeo:STATe?
```

## Arguments

OFF or 0 turns the video bandwidth filter off.

ON or 1 turns the video bandwidth filter on.

## Examples

SENSe:SEM:OFFSet3:BANDwidth:VIDeo:STATe ON turns on the video bandwidth filter for row 3 in the table.

## [SENSe]:SEM:OFFSet<x>:FREQuency:START

Sets or queries the start frequency for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:FREQuency:START <value>
```



```
[SENSe]:SEM:OFFSet<x>:FREQuency:START?
```

## Arguments

<value>::<NRf> specifies the start frequency.

## Examples

SENSe:SEM:OFFSet1:FREQuency:START 2.876MHZ sets the start frequency in row 1 to 2.876 MHz.

## [SENSe]:SEM:OFFSet<x>:FREQuency:STOP

Sets or queries the stop frequency for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:FREQuency:STOP <value>
```

```
[SENSe]:SEM:OFFSet<x>:FREQuency:STOP?
```

## Arguments

<value>::<NRf> specifies the stop frequency.

## Examples

SENSe:SEM:OFFSet1:FREQuency:STOP 3.485MHZ ssets the stop frequency in row 1 to 3.485 MHz.

## [SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:SAME

Sets or queries the state of the limits for the absolute start and stop settings for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:SAME { OFF | ON | 0 | 1 }
```

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:SAME?
```

## Arguments

OFF or 0 indicates the start and stop limits are different.

ON or 1 indicates the state and stop limits are the same.

## Examples

`SENSe:SEM:OFFSet2:LIMIT:ABSolute:SAME ON` sets the absolute start and stop limits for row 2 to the same values.

## [SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:START

Sets or queries the absolute power start limit for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:START <value>
```

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:START?
```

## Arguments

<value>::<NRf> specifies the power in dBm.

## Examples

`SENSe:SEM:OFFSet1:LIMIT:ABSolute:START -71.10` sets the absolute start power limit for row 1 to -71.10 dBm.

## [SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:STOP

Sets or queries the absolute power stop limit for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:STOP <value>
```

```
[SENSe]:SEM:OFFSet<x>:LIMIT:ABSolute:STOP?
```

## Arguments

<value> : : <NRf> specifies the power in dBm.

## Examples

SENSe:SEM:OFFSet<x>:LIMIT:ABSolute:STOP -55.80 sets the absolute stop power limit for row 1 to -55.80 dBm.

## [SENSe]:SEM:OFFSet<x>:LIMIT:MASK

Sets or queries the limit mask setting for the specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:MASK { ABS | REL | AND | OR }
```

```
[SENSe]:SEM:OFFSet<x>:LIMIT:MASK?
```

## Arguments

ABS specifies the limit mask is set to Absolute.

REL specifies the limit mask is set to Relative

AND specifies the limit mask is set to Absolute AND Relative.

OR specifies the limit mask is set to Absolute OR Relative.

## Examples

SENSe:SEM:OFFSet3:LIMIT:MASK REL sets the limit mask for row 3 to Relative.

## [SENSe]:SEM:OFFSet<x>:LIMIT:RELative:SAME

Sets or queries the state of the limits for the relative start and stop settings for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:SAME { OFF | ON | 0 | 1 }
```

[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:SAME?

### Arguments

OFF or 0 indicates the start and stop limits are different.

ON or 1 indicates the state and stop limits are the same.

### Examples

SENSe:SEM:OFFSet2:LIMIT:RELative:SAME ON sets the relative start and stop limits for row 2 to the same values.

## [SENSe]:SEM:OFFSet<x>:LIMIT:RELative:START

Sets or queries the relative power start limit for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Sense commands

### Syntax

[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:START <value>

[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:START?

### Arguments

<value>::<NRf> specifies the power in dB.

### Examples

SENSe:SEM:OFFSet1:LIMIT:RELative:START -35.5 sets the relative start power limit for row 1 to -35.5 dB.

## [SENSe]:SEM:OFFSet<x>:LIMIT:RELative:STOP

Sets or queries the relative power stop limit for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Sense commands

### Syntax

[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:STOP <value>

```
[SENSe]:SEM:OFFSet<x>:LIMIT:RELative:STOP?
```

## Arguments

<value>::<NRf> specifies the power in dB.

## Examples

SENSe:SEM:OFFSet1:LIMIT:RELative:STOP -35.5 sets the relative stop power limit for row 1 to 35.5 dB.

## [SENSe]:SEM:OFFSet<x>:LIMIT:SIDE

Sets or queries the side of the offset limit for specified row in the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:LIMIT:SIDE { LEFT | RIGHT | BOTH }
```

```
[SENSe]:SEM:OFFSet<x>:LIMIT:SIDE?
```

## Arguments

LEFT specifies the left offset limit.

RIGHT specifies the right offset limit.

BOTH specifies both sides of the offset limit.

## Examples

SENSe:SEM:OFFSet1:LIMIT:SIDE BOTH specifies the measurement uses both sides of the offset in row 1.

## [SENSe]:SEM:OFFSet<x>:STATe (No Query Form)

Determines whether to display the specified row of the Offsets & Limits table. The parameter <x> represents the row in the Offsets & Limits table.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:OFFSet<x>:STATe { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 turns off the display of the row in the measurement view.

ON or 1 displays the row in the measurement view.

## Examples

```
SENSe:SEM:OFFSet2:STATe OFF the display of row 2 in the measurement view.
```

## [SENSe]:SEM:POINTs:COUNT

Sets or queries the maximum number of trace points per range.



**Note:** *In Swept operation this is the number of trace points per range. In Real-time operation it is the number of trace points in the entire trace.*

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:POINTs:COUNT <value>
```

```
[SENSe]:SEM:POINTs:COUNT?
```

## Arguments

None

## Returns

P<NR1> number of points per range.

## Examples

```
SENSe:SEM:POINTs:COUNT? might return P8001 indicating a maximum of 8001 trace points.
```

## [SENSe]:SEM:PRESet (No Query Form)

Defines the standard protocol settings for the Spectral Emissions Mask measurement.



**Note:** This is an obsolete command.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:PRESet { "3GPP Downlink" | "WLAN 802.11g" | "WiMax 802.16 (ETSI)" }
```

## Arguments

"3GPP Downlink" specifies the 3GPP Downlink standard.

"WLAN 802.11g" specifies the IEEE 802.11 standard for WLAN communications.

"WiMax 802.16 (ETSI)" specifies the WiMAX 802.16 (ETSI) standard. Quotes must be included with the arguments.

## Examples

```
SENSe:SEM:PRESET WLAN (802.11g) presets the SEM measurement to the IEEE 802.11 standard for WLAN communications.
```

## [SENSe]:SEM:REference:AUTO

Determines whether to automatically update the Power Reference in the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:REference:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SEM:REference:AUTO?
```

## Arguments

OFF or 0 turns off the automatic update.

ON or 1 automatically updates the reference power.

## Examples

```
SENSe:SEM:REFERENCE:AUTO ON automatically updates the reference power in the measurement.
```

## [SENSe]:SEM:REference:DETection

Sets or returns the reference channel detection setting.



**Note:** In Real-time operation the offset and reference detection settings are forced equal.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:REFEreNce:DETEction { AVERage | PEAK }  
[SENSe]:SEM:REFEreNce:DETEction?
```

## Arguments

AVERage indicates the reference channel detection setting is Average (VRMS).

PEAK indicates the reference channel detection setting is the positive peak.

## Examples

SENSe:SEM:REFERENCE:DETECTION AVER sets the reference channel detection setting to Average RMS.

## [SENSe]:SEM:REFEreNce[:MAGNitude]

Sets or queries the reference power magnitude.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:REFEreNce[:MAGNitude] <Value>  
[SENSe]:SEM:REFEreNce[:MAGNitude]?
```

## Arguments

<value>::<NRf> specifies the magnitude in dBm.

## Examples

SENSe:SEM:REFERENCE[:MAGNITUDE]? might return a magnitude of -51.057.

## [SENSe]:SEM:RRCRolloff

Sets or queries the filter parameter (roll-off ratio) for the Root Raised Cosine filter.



**Note:** Applies in Swept operation only.



## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:RRCRolloff <value>
```

```
[SENSe]:SEM:RRCRolloff?
```

## Arguments

<value>: : <NRf> specifies the filter parameter.

Range: 0.001 to 1, 0.001 step.

## Examples

SENSe:SEM:RRCROLLOFF 0.22 sets the filter parameter to 0.22.

## [SENSe]:SEM:STEP:AUTO

Determines whether to automatically update the center frequency increment/decrement step size in the Spectral Emissions Mask measurement. The Step value controls the magnitude of each step when using the front panel controls or arrow keys on the keyboard.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Sense commands

## Syntax

```
[SENSe]:SEM:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SEM:STEP:AUTO?
```

## Related Commands

```
[SENSe]:SEM:STEP[:MAGNitude]
```

## Arguments

OFF or 0 turns off the automatic update.

ON or 1 automatically updates center frequency.

## Examples

SENSe:SEM:STEP:AUTO ON automatically updates the center frequency.

## **[SENSe]:SEM:STEP[:MAGNitude]**

Sets or queries the step magnitude for the center frequency in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SEM:STEP[:MAGNitude] <value>
```

```
[SENSe]:SEM:STEP[:MAGNitude]?
```

### **Related Commands**

```
[SENSe]:SEM:STEP:AUTO
```

### **Arguments**

<value>::<Nrf> specifies the step setting.

### **Examples**

```
SENSe:SEM:STEP[:MAGNITUDE] 4MHZ sets the step magnitude to 4 MHz.
```

## **[SENSe]:SEM:WIDTH:DEFinition**

Sets or queries the offset definition of the measurement type in the Spectral Emissions Mask measurement.

### **Conditions**

Measurement views: Spectral Emissions Mask

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SEM:WIDTH:DEFinition {C2C | C2E | E2C | E2E}
```

```
[SENSe]:SEM:WIDTH:DEFinition?
```

### **Arguments**

C2C defines the offset from the reference frequency center to the offset center.

C2E defines the offset from the reference frequency center to the offset edge.

E2C defines the offset from the reference frequency edge to the offset center.

E2E defines the offset from the reference frequency edge to the offset edge.

## Examples

SENSe:SEM:WIDTH:DEFINITION C2C specifies the offset limit from the center of the reference frequency to the center of the offset.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets SENSe:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution] <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution] ?
```

## Related Commands

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:RESolution:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the RBW. Range: 1 Hz to 10 MHz.

## Returns

<NRf> specifies the RBW.

## Examples

```
[SENSe]:SGRam:MEASview1:BANDwidth:RESolution 200kHz sets the RBW to 200 kHz.
```

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:ACTual (Query Only)

Queries the actual resolution bandwidth (RBW) in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> The actual RBW in Hz.

## Examples

SENSe:SGRam:MEASview1:BANDwidth:RESolution:ACTual? might return 299.624E+3, indicating that the actual RBW is 299.624 kHz.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 specifies that the resolution bandwidth is set manually using the [SENSe]:DPX:{BANDwidth|BWIDth}:ACTual? command.

ON or 1 specifies that the resolution bandwidth is set automatically.

## Returns

{ OFF | ON | 0 | 1 }

## Examples

SENSe:SGRam:MEASview1:BANDwidth:RESolution:AUTO ON sets the resolution bandwidth automatically.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE

Determines whether to enable or disable the RBW processing in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE { OFF | ON | 0 | 1 }
```

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 disables the RBW processing. You can select the FFT window using the `SENSe:SGRam[:MEASview<y>]:FFT:WINDow` command.

ON or 1 enables the RBW processing. Refer to the `SENSe:SGRam:MEASview<y>:BANDwidth:RESolution` command to set the RBW.

### Returns

```
{ OFF | ON | 0 | 1 }
```

### Examples

`SENSe:SGRam:MEASview1:BANDwidth:RESolution:MODE ON` enables the RBW processing.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets `SENSe:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe` OFF.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo?
```

## Related Commands

`SENSe:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the VBW.

Range: Current RBW/104 (1 Hz minimum) to Current RBW.

## Returns

<NRf> specifies the VBW.

## Examples

`SENSe:SGRam:MEASview1:BANDwidth:VIDeo 200kHz` sets the VBW to 200 kHz.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the spectrogram measurement.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

`[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON | 0 | 1 }`

`[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

## Returns

{ OFF | ON | 0 | 1 }

## Examples

`SENSe:SGRam:MEASview1:BANDwidth:VIDeo:STATe ON` enables the VBW.

## [SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:OPTimization

Sets or queries the method of optimizing the gain and input bandwidth in the Spectrogram measurement.

### Conditions

Measurement view: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:OPTimization { BEST | AUTO |
MAXDynrange | MINNoise | MINTime }
```

```
[SENSe]:SGRam[:MEASview<y>]:{BANDwidth|BWIDth}:OPTimization?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

BEST - BEST (Best for multiple displays) causes the instrument to evaluate all the open displays and select an Acquisition Bandwidth wide enough to satisfy as many measurements as possible

AUTO - AUTO sets the gain and input bandwidth for an optimized trade-off of the competing performance choices.

MAXDynrange - MAXDynrange optimizes the gain and input bandwidth to maximize the dynamic range.

MINNoise - MINNoise optimizes the gain and input bandwidth to minimize noise.

MINTime - MINTime optimizes the gain and input bandwidth to minimize sweep time.

### Returns

```
{ BEST | AUTO | MAXDynrange | MINNoise | MINTime }
```

### Examples

SENSe:SGRam:MEASview1:BANDwidth:OPTimization? might return AUTO which optimizes automatically the gain and input.

## [SENSe]:SGRam[:MEASview<y>]:COLor

Sets or queries the color palette of three-dimensional graphs.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:COLor { RED | GREen | BLUe | CYAN | BCYan| YELLow |
MAGenta | GRAY | TEMPerature | SPECtral }
```

```
[SENSe]:SGRam[:MEASview<y>]:COLor?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

The following table lists the arguments.

**Table 53: Color palette for spectrogram**

Argument	Palette
RED	Red
GREen	Green
BLUe	Blue
CYAN	Cyan
BCYan	Binary cyan
YELLow	Yellow
MAGenta	Magenta
GRAY	Gray
TEMPerature	Temperature
SPECtral	Spectral

## Returns

```
{ RED | GREen | BLUe | CYAN | BCYan| YELLow | MAGenta | GRAY | TEMPerature | SPECtral }
```

## Examples

SENSe:SGRam:MEASview1:COLor TEMPerature selects the temperature color palette.

## [SENSe]:SGRam[:MEASview<y>]:COLor:MAXimum

Sets or queries the maximum value of the color axis in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:COLor:MAXimum <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:COLor:MAXimum?
```



## Related Commands

`SENSe:SGRam[:MEASview<y>]:COLor:MINimum`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the maximum value of the color axis.

Range: -100 to +100 dBm.

## Returns

<NRf> specifies the maximum value of the color axis.

## Examples

`SENSe:SGRam:MEASview1:COLor:MAXimum 10` sets the maximum value of the color axis to 10 dBm.

## [SENSe]:SGRam[:MEASview<y>]:COLor:MINimum

Sets or queries the minimum value of the color axis in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

`[SENSe]:SGRam[:MEASview<y>]:COLor:MINimum <value>`

`[SENSe]:SGRam[:MEASview<y>]:COLor:MINimum?`

## Related Commands

`SENSe:SGRam[:MEASview<y>]:COLor:MAXimum`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the minimum value of the color axis.

Range: -100 to +100 dBm.

## Returns

<NRf> specifies the minimum value of the color axis.

## Examples

`SENSe:SGRam:MEASview1:COLor:MINimum 10` sets the minimum value of the color axis to 10 dBm.

## [SENSe]:SGRam[:MEASview<y>]:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

### Conditions

Measurement view: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:CLEar:RESults
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

`SENSe:SGRam:MEASview1:CLEar:RESults` clears the results of spectrogram and restarts multi-trace functions.

## [SENSe]:SGRam[:MEASview<y>]:FFT:WINDow

Sets or queries the FFT window in the spectrogram. This command is equivalent to `SENSe:SGRam[:MEASview<y>]:FILTer:SHAPE`.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FFT:WINDow { KAISer | MIL6db | CISPr | BH4B | UNIFORM | FLATtop | HANNing }
```

```
[SENSe]:SGRam[:MEASview<y>]:FFT:WINDow?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

`KAISer` selects the Kaiser (RBW) window.

MIL6db selects the -6 dB RBW (MIL) window.

CISPr selects the CISPR window.

BH4B selects the Blackman-Harris 4B type window.

UNIFORM selects the uniform window.

FLATtop selects the flat-top window.

HANNing selects the Hanning window.

## Returns

{ KAISer | MIL6db | CISPr | BH4B | UNIFORM | FLATtop | HANNing }

## Examples

SENSe:SGRam:MEASview1:FFT:WINDow HANNing selects the Hanning window.

## [SENSe]:SGRam[:MEASview<y>]:FILTer[:SHAPE]

Sets or queries the filter shape in the spectrogram. This command is equivalent to  
SENSe:SGRam[:MEASview<y>]:FFT:WINDow.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

[SENSe]:SGRam[:MEASview<y>]:FILTer[:SHAPE] { KAISer | MIL6db | CISPr | BH4B | UNIFORM | FLATtop | HANNing }

[SENSe]:SGRam[:MEASview<y>]:FILTer[:SHAPE]?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

KAISer selects the Kaiser (RBW) window.

MIL6db selects the -6 dB RBW (MIL) window.

CISPr selects the CISPR window.

BH4B selects the Blackman-Harris 4B type window.

UNIFORM selects the uniform window.

FLATtop selects the flat-top window.

HANNing selects the Hanning window.

## Returns

{ KAISer | MIL6db | CISPr | BH4B | UNIFORM | FLATtop | HANNing }

## Examples

SENSe:SGRam:MEASview1:FILTer:SHApe HANNing selects the Hanning window.

## [SENSe]:SGRam[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the spectrogram.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships: (start frequency) = (center frequency) - (span)/2 and (stop frequency) = (center frequency) + (span)/2.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

[SENSe]:SGRam[:MEASview<y>]:FREQuency:CENTer <value>

[SENSe]:SGRam[:MEASview<y>]:FREQuency:CENTer?

## Related Commands

[SENSe]:SGRam[:MEASview<y>]:FREQuency:START

[SENSe]:SGRam[:MEASview<y>]:FREQuency:STOP

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Returns

<NRf> specifies the center frequency.

## Examples

SENSe:SGRam:MEASview1:FREQuency:CENTer 7.5GHz sets the center frequency to 7.5 GHz.

## [SENSe]:SGRam[:MEASview<y>]:FREQUENCY:SPAN

Sets or queries the frequency span in the spectrogram.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:SPAN <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:SPAN?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASView keyword is optional by default y is 1.

<value> : : <NRf> specifies the frequency span.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

### Returns

<NRf> specifies the frequency span.

### Examples

SENSe:SGRam:MEASview1:FREQUENCY:SPAN 20MHz sets the span to 20 MHz.

## [SENSe]:SGRam[:MEASview<y>]:FREQUENCY:SPAN:BANDwidth[:RESolution]:RATio

Sets or queries the ratio of span to RBW (Resolution Bandwidth) in the spectrogram. This command is valid when SENSe:SGRam[:MEASview<y>]:BANDwidth:RESolution:AUTO is set to On.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio <value>  
[SENSe]:SGRam[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the ratio of span to RBW. Range: 20 to 1000.

Programming a specified ratio sets the RBW (= span/ratio), which is rounded down to the nearest valid value.

## Returns

<NRf> specifies the ratio of span to RBW.

## Examples

SENSe:SGRam:MEASview1:FREQuency:SPAN:BANDwidth:RESolution:RATio 200 sets the ratio to 200, setting the RBW to 200 kHz for the span of 40 MHz.

## [SENSe]:SGRam[:MEASview<y>]:FREQuency:SPAN:MAXimum (No Query Form)

Sets the frequency range to the maximum real-time span in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FREQuency:SPAN:MAXimum
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

SENSe:SGRam:MEASview1:FREQuency:SPAN:MAXimum sets the frequency range to the maximum real-time span.

## [SENSe]:SGRam[:MEASview<y>]:FREQuency:START

Sets or queries the measurement start frequency (left edge of the graph) in the spectrogram.

The center, start and stop frequencies are set interlocking each other. Refer to the SENSe:SGRam[:MEASview<y>]:FREQuency:CENTercommand.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:START <value>
```

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:START?
```

## Related Commands

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STOP
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> is the measurement start frequency.

Range: (center frequency) ± (span)/2.

## Returns

<NRf> specifies the measurement start frequency.

## Examples

`SENSE:SGRam:MEASview1:FREQUENCY:START 6.95GHz` sets the start frequency to 6.95 GHz.

## [SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STEP

Sets or queries the frequency step size. Programming a specified step size sets

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STEP:AUTO OFF.
```

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STEP <value>
```

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STEP?
```

## Related Commands

```
[SENSE]:SGRam[:MEASview<y>]:FREQUENCY:STEP:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Returns

<NRf> specifies the frequency step size.

## Examples

`SENSe:SGRam:MEASview1:FREQUENCY:STEP 1kHz` sets the frequency step size to 1 kHz.

## [SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STEP:AUTO

Determines whether to set the frequency step size automatically or manually in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STEP:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STEP:AUTO?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 specifies that the frequency step size is set manually using the `SENSe:SGRam[:MEASview<y>]:FREQUENCY:STEPsize` command.

ON or 1 specifies that the frequency step size is set automatically.

## Returns

```
{ OFF | ON | 0 | 1 }
```

## Examples

`SENSe:SGRam:MEASview1:FREQUENCY:STEP:AUTO ON` specifies that the frequency step size is set automatically.



## [SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STOP

Sets or queries the measurement stop frequency (right edge of the graph) in the spectrogram.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SGRam[:MEASview<y>]:FREQUENCY:CENTer command.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STOP <value>
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:STOP?
```

### Related Commands

```
[SENSe]:SGRam[:MEASview<y>]:FREQUENCY:START
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> is the measurement stop frequency.

Range: (center frequency) ± (span)/2.

### Returns

<NRf> specifies the measurement stop frequency.

### Examples

```
SENSe:SGRam:MEASview1:FREQUENCY:STOP 7.05GHz sets the stop frequency to 7.05 GHz.
```

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:MODE

Sets or queries the vertical mode.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:MODE {NORMal | MONitor}
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:MODE?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

NORMAL indicates the instrument is set to normal mode.

MONitor indicates the instrument is set to Spectrum Monitor mode.

### Returns

{NORMAL | MONitor}

### Examples

SENSe:SGRam:MEASview1:TIME:SCALE:MODE MONITOR set the instrument to Spectrum Monitor mode.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap

Sets or queries the Overlap. Note that changing this value may change the time per division value (DISPlay:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision).

### Conditions

Measurement view: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf> specifies the overlap.

### Returns

<NRf> specifies the overlap.

### Examples

SENSe:SGRam:MEASview1:TIME:SCALE:OVERlap 1.0E-1 sets the overlap to 0.1.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap:PERCent

Sets or queries the Overlap in percent. Note that changing this value may change the time per division  
SENSe:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision value.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap:PERCent <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:OVERlap:PERCent?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : :<NRf> specifies the overlap in percent.

### Returns

<NRf> specifies the overlap in percent.

### Examples

SENSe:SGRam:MEASview1:TIME:SCALE:OVERlap:PERCent 1.0E1 sets the overlap to 10%.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision

Sets or queries the vertical Time/division value. Note that changing this value may change the  
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent value.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:DIVision?
```

### Related Commands

```
DISPlay:SGRam[:MEASview<y>]:TIME:OVERlap:PERCent
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NRf> specifies the time per division in seconds.

## Returns

<NRf> specifies the overlap in percent.

## Examples

SENSe:SGRam:MEASview1:TIME:SCALE:PER:DIVision? might return 1.116000000E-3.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:ZOOM

Sets or queries the zoom scale in the scale tab of spectrogram.

## Conditions

Measurement view: Spectrogram

## Group

Sense commands

## Syntax

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:ZOOM <value>

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:ZOOM?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf> specifies the zoom scale value.

## Returns

<NRf> specifies the overlap.

## Examples

SENSe:SGRam:MEASview1:TIME:SCALE:ZOOM2 sets the zoom scale to 2.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:AUTO

Sets the scale to autoscale in the spectrogram.

## Conditions

Measurement view: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

None

## Examples

SENSe:SGRam:MEASview1:TIME:SCALe:AUTO sets the scale to auto.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:START

Sets or queries the start time in the spectrogram.

## Conditions

Measurement view: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:START <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALe]:START?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> = <NRf> specifies the start time in seconds.

## Returns

<NRf> specifies the start time in seconds.

## Examples

SENSe:SGRam:MEASview1:TIME:SCALe:START? might return 1E-6 s.

## **[SENSe]:SGRam[:MEASview<y>]:SELEcted:SOURce**

Returns the Source number of specified Spectrogram trace.

### **Conditions**

Measurement view: Spectrogram

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SGRam[:MEASview<y>]:SELEcted:SOURce {"Source 1"|"Source 2"|"Source 3"|"Source 4"}
```

```
[SENSe]:SGRam[:MEASview<y>]:SELEcted:SOURce?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### **Returns**

```
{"Source 1"|"Source 2"|"Source 3"|"Source 4"}
```

### **Examples**

SENSe:SGRam:MEASview1:SELEcted:SOURce? might return Source 2 for the trace of Spectrogram measurement view 1.

## **[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:MINutes**

Sets or queries the Spectrum Monitor mode Time/Update value for minutes.

### **Conditions**

Measurement views: Spectrogram

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:MINutes <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:MINutes?
```

### **Arguments**

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> sets the time/update value in minutes.

### Returns

<NRf> specifies the time/update value in minutes.

### Examples

SENSe:SGRam:MEASview1:TIME:SCALE:PER:UPDate:MINutes 1E1 sets the Time/update minutes value to 10.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:SEConds

Sets or queries the Spectrum Monitor mode Time/Update value for seconds.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

### Syntax

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:SEConds <value>

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:PER:UPDate:SEConds?

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> sets the time/update value in seconds.

### Returns

<NRf> specifies the time/update value in seconds.

### Examples

SENSe:SGRam:MEASview1:TIME:SCALE:PER:UPDate:SEConds 1E1 sets the Time/update seconds value to 10.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:SPECTrums:PERLine (Query Only)

Queries the Spectrums/Line value.

### Conditions

Measurement views: Spectrogram

### Group

Sense commands

## Syntax

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:SPECTrums:PERLine?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

<NRf> specifies Spectrum/Line value.

## Examples

SENSe:SGRam:MEASview1:TIME:SCALE:SPECTrums:PERLine? might return a 1.0000000000 as a response.

## [SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:START:DIVisions

Sets or queries the Vertical offset or Position in divisions.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:START:DIVisions <value>

[SENSe]:SGRam[:MEASview<y>]:TIME[:SCALE]:START:DIVisions?

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> sets the vertical offset value in divisions. Range is -130,000 to +130,000.

## Returns

<NRf> specifies vertical offset value in divisions.

## Examples

SENSe:SGRam:MEASview1:TIME:SCALE:START:DIVisions 10 sets the vertical offset to 10 divisions

## [SENSe]:SGRam[:MEASview<y>]:WATerfall:DIRection

Sets or queries the direction of the Waterfall (3-D spectrogram) display.



## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:DIRection { NW | NE }
```

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:DIRection?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

NW sets the display direction to northwest.

NE sets the display direction to northeast.

## Returns

{ NW | NE }

## Examples

SENSe:SGRam:MEASview1:WATerfall:DIRection NW sets the waterfall display direction to northwest.

## [SENSe]:SGRam[:MEASview<y>]:WATerfall:ENABLE

Sets or queries the Waterfall (3-D spectrogram) display.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:ENABLE { OFF | ON | 0 | 1 }
```

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:ENABLE?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 disables the Waterfall display.

ON or 1 enables the Waterfall display.

## Returns

{ OFF | ON | 0 | 1 }

## Examples

SENSe:SGRam:MEASview1:WATerfall:ENABle ON enables the 3-D Waterfall display.

## [SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALE]:AUTO (No Query Form)

Rescales the height axis automatically to fit the waveform to the screen in the 3-D Waterfall display.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALE]:AUTO
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Examples

SENSe:SGRam:MEASview1:WATerfall:Y:SCALE:AUTO rescales the height scale automatically to fit the waveform to the 3-D Waterfall display.

## [SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALE]:OFFSet

Sets or queries the height offset (the value at the top edge of the vertical axis) of the 3-D Waterfall display.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALE]:OFFSet <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:WATerfall:Y[:SCALE]:OFFSet?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NRf> specifies the height (vertical) offset. Range: -270 to +150 dBm.

## Returns

<NRf> specifies the height (vertical) offset.

## Examples

SENSe:SGRam:MEASview1:WATERfall:Y:SCALE:OFFSET 12.5 sets the offset to 12.5 dB in the 3-D Waterfall display.

## [SENSe]:SGRam[:MEASview<y>]:WATERfall:Y[:SCALE]:RESet (No Query Form)

Resets the height scale of the 3-D Waterfall display to the default values: Height position= 0 dB and Height scale = 100 dB

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATERfall:Y[:SCALE]:RESet
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASView keyword is optional by default y is 1.

## Examples

SENSe:SGRam:MEASview1:WATERfall:Y:SCALE:RESEt rescales the height scale automatically to fit the waveform to the 3-D Waterfall display.

## [SENSe]:SGRam[:MEASview<y>]:WATERfall:Y[:SCALE]

Sets or queries the height (vertical) range of the Waterfall (3-D spectrogram) display.

## Conditions

Measurement views: Spectrogram

## Group

Sense commands

## Syntax

```
[SENSe]:SGRam[:MEASview<y>]:WATERfall:Y[:SCALE] <value>
```

```
[SENSe]:SGRam[:MEASview<y>]:WATERfall:Y[:SCALE]?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value> : : <NRf> specifies the vertical range. Range: 0.1 to 200 dB. The amplitude unit is set by the [SENSe]:POWer:UNITs command.

You can omit the unit in the argument. When you include the unit in the argument, only dB is available. For the amplitude units Watts, Volts, and Amps, omit the unit in the argument.

## Returns

<NRf> specifies the vertical range.

## Examples

SENSe:SGRam:MEASview1:WATerfall:Y:SCALe 100dB sets the height range to 100 dB in the 3-D Waterfall display.

## [SENSe]:SIGNAlpath:PRlority

Sets or queries the signal path priority for the measurements.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNAlpath:PRIority { RF | LF }
```

```
[SENSe]:SIGNAlpath:PRIority?
```

## Arguments

RF sets the signal path to high frequency range.

LF sets the signal path to low frequency range.

## Examples

SENSe:SIGNALPATH:PRIORITY sets the low frequency signal path when measurements are completely within LF frequency range.

## [SENSe]:SIGNAlvu:ACQuisition:CONTrol:HORIZontal

Sets or queries the other acquisition or Horizontal settings. When the checkbox is checked, the SignalVu controls the Horizontal settings of oscilloscope.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:HORizontal?  
[SENSe]:SIGNalvu:ACQuisition:CONTRol:HORizontal { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Horizontal settings checkbox.

ON or 1 enables Horizontal settings checkbox.

## Returns

0 disables Horizontal settings checkbox.

1 enables Horizontal settings checkbox.

## Examples

`SENSe:SIGNalvu:ACQuisition:CONTRol:HORizontal OFF` disables the Horizontal settings from the SignalVu.

## [SENSe]:SIGNalvu:ACQuisition:CONTRol:SAMPLE:RATE

Sets or queries the Sample rate, when the checkbox is checked, the SignalVu controls the Sample rate of the oscilloscope.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:SAMPLE:RATE?  
[SENSe]:SIGNalvu:ACQuisition:CONTRol:SAMPLE:RATE { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Sample rate checkbox.

ON or 1 enables Sample rate checkbox.

## Returns

0 disables Sample rate checkbox.

1 enables Sample rate checkbox.

## Examples

`SENSe:SIGNalvu:ACQuisition:CONTRol:SAMPLe:RATE OFF` disables the Sample rate from the SignalVu.

## [SENSe]:SIGNalvu:ACQuisition:CONTRol:SOURce<x>:STATe

Sets or queries the Source checkbox state of measurement source for RF,I & Q, Diff I & Q data type

### Conditions

Should be connected to MSO/LPD to Set or Query.

### Group

Sense commands

### Syntax

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:SOURce<x>:STATe?
```

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:SOURce<x>:STATe { OFF | ON | 0 | 1 }
```

### Arguments

<x>=1 to 8 representing Range of source

OFF or 0 disables Source State.

ON or 1 enables Source State.

### Returns

0 disables Source State.

1 enables Source State.

## Examples

`SENSe:SIGNalvu:ACQuisition:CONTRol:SOURce1:STATe OFF` disables Source State.

## [SENSe]:SIGNalvu:ACQuisition:CONTRol:TRIGger:POSition

Sets or queries the Trigger position, when the checkbox is checked, the SignalVu controls the Trigger position of oscilloscope.

### Conditions

Should be connected to MSO/LPD/SX to set or query.

### Group

Sense commands

### Syntax

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:TRIGger:POSition?
```

```
[SENSe]:SIGNalvu:ACQuisition:CONTRol:TRIGger:POSition { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Trigger position checkbox.

ON or 1 enables Trigger position checkbox.

## Returns

0 disables Trigger position checkbox.

1 enables Trigger position checkbox.

## Examples

`SENSe:SIGNALvu:ACQuisition:CONTRol:TRIGger:POSition OFF` disables the Trigger position from the SignalVu.

## [SENSe]:SIGNALvu:ACQuisition:CONTRol:VERTical

Sets or queries the Vertical settings, when the checkbox is checked, the SignalVu controls the vertical settings of the oscilloscope.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNALvu:ACQuisition:CONTRol:VERTical?
```

```
[SENSe]:SIGNALvu:ACQuisition:CONTRol:VERTical { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 disables Vertical settings checkbox.

ON or 1 enables Vertical settings checkbox.

## Returns

0 disables Vertical settings checkbox.

1 enables Vertical settings checkbox.

## Examples

`SENSe:SIGNALvu:ACQuisition:CONTRol:VERTical OFF` disables Vertical settings from the SignalVu.

## [SENSe]:SIGNALvu:ACQuisition:DIgitizer:SAMPle:RATE

Sets or queries the oscilloscope Sample rate from the Acquisition control panel oscilloscope settings of the SignalVu.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE?
```

```
[SENSe]:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE
```

## Arguments

<value>::<NRf> specifies the sample rate.

Range: 3.125 S/s to 100 GS/s.

## Returns

<NRf> specifies the sample rate.

## Examples

SENSe:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE 50E+9, sets the oscilloscope sample rate to 50 GS/s.

## [SENSe]:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE:AUTO

Determines whether to set the oscilloscope Sample rate automatically or manually from the Acquisition control panel oscilloscope settings of the SignalVu.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE:AUTO?
```

```
[SENSe]:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE:AUTO { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 specifies the oscilloscope sample rate is set manually.

ON or 1 specifies the oscilloscope sample rate is set automatically.

## Examples

SENSe:SIGNalvu:ACQuisition:DIGitizer:SAMPlE:RATE:AUTO ON sets the oscilloscope Sample rate automatically.



## [SENSe]:SIGNalvu:ACQuisition:GLOBal:REFlevel

Sets or queries the Global Reference level.

### Conditions

Should be connected to MSO/LPD to Set or Query.

### Group

Sense commands

### Syntax

```
[SENSe]:SIGNalvu:ACQuisition:GLOBal:REFlevel? (Query)
[SENSe]:SIGNalvu:ACQuisition:GLOBal:REFlevel <value>
```

### Arguments

<value>::<NRf> specifies the global reference level

Range: {-130 to 30} in dBm

### Returns

<NRf>

### Examples

SENSe:SIGNalvu:ACQuisition:GLOBal:REFlevel? might return 5, which indicates the global ref level value is 5.

## [SENSe]:SIGNalvu:ACQuisition:INPuttype:REFlevel

Sets or queries the Reference Level input type.

### Conditions

Should be connected to MSO/LPD to Set or Query.

### Group

Sense commands

### Syntax

```
[SENSe]:SIGNalvu:ACQuisition:INPuttype:REFlevel? (Query)
[SENSe]:SIGNalvu:ACQuisition:INPuttype:REFlevel {PERChannel | PERSource | GLOBal}
```

### Arguments

PERChannel sets the Input Reflevel to Per Channel.

PERSource sets the Input Reflevel to Per Source.

GLOBal sets the Input Reflevel to Global.

## Examples

`SENSe:SIGNalvu:ACQquisition:INPuttype:REFlevel PERChannel` sets input relevel to per channel.

## **[SENSe]:SIGNalvu:ACQquisition:IQ:CALibrate (No query form)**

Starts the IQ calibration.

## Conditions

Should be connected to MSO/LPD/SX.

## Group

Sense commands

## Syntax

`SENSe:SIGNalvu:ACQquisition:IQ:CALibrate`

## Examples

`SENSe:SIGNalvu:ACQquisition:IQ:CALibrate` starts the IQ calibration.

## **[ :SENSe]:SIGNalvu:ACQquisition:NULL:IQ:OFFSet:CLEAR**

Use this command to clear Null IQ offset values that were calculated using Calibrate IQ.

## Related Commands

`SIGNalvu:ACQquisition:SOURce:CALIBrate:IQ`

## Conditions

Should be connected to MSO/LPD/SX to query.

## Group

Sense commands

## Syntax

`[ :SENSe]:SIGNalvu:ACQquisition:NULL:IQ:OFFSet:CLEAR`

## Examples

`:SENSe:SIGNalvu:ACQquisition:NULL:IQ:OFFSet:CLEAR`

## **[ :SENSe ]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:COUPling**

Sets or queries the per channel coupling of the measurement. The parameter <x>=1 to 8 represents the channel index.

### **Conditions**

Should be connected to MSO/LPD set or query.

This cmd is query only in case of SX.

### **Group**

Sense commands

### **Syntax**

```
:SENSe:SIGNAlvu:ACQuisition:PER:CHANnel<x>:COUPling {AC | DC}
```

```
:SENSe:SIGNAlvu:ACQuisition:PER:CHANnel<x>:COUPling? (Query)
```

### **Argument**

AC sets the specified channel to AC coupling.

DC sets the specified channel to DC coupling.

### **Examples**

```
SENSe:SIGNAlvu:ACQuisition:PER:CHANnel1:COUPling AC, sets coupling of channel 1 to AC.
```

## **[ :SENSe ]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:DESkew**

Sets or queries the per channel Deskew of the measurement. The parameter <x>=1 to 8 represents the channel index.

### **Conditions**

Should be connected to MSO/LPD/SX to set or query.

### **Group**

Sense commands

### **Syntax**

```
[ :SENSe ]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:DESkew <value>
```

```
[ :SENSe ]:SIGNAlvu:ACQuisition:PER:CHANnel <x>:DESkew? (Query)
```

### **Argument**

<value>::<NRf> specifies the deskew.

### **Returns**

<NRf>

## Examples

SENSe:SIGNalvu:ACQuisition:PER:CHANnel:DESkew:RF 125.000000000E-9, sets the deskew of channel measurement to 125 ns.

## [[:SENSe]:SIGNalvu:ACQuisition:PER:CHANnel<x>:EXTattn

Sets or queries the per channel External attenuation of the measurement. The parameter <x>=1 to 8 represents the channel index.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

[[:SENSe]:SIGNalvu:ACQuisition:PER:CHANnel<x>:EXTattn <value>

[[:SENSe]:SIGNalvu:ACQuisition:PER:CHANnel<x>:EXTattn? (Query)

## Argument

<value>::<NRf> specifies the external attenuation.

## Returns

<NRf>

## Examples

SENSe:SIGNalvu:ACQuisition:PER:CHANnel1:EXTattn:RF 30.000000000 sets external attenuation of channel measurement to 30 dB.

## SENSe:SIGNalvu:ACQuisition:PER:CHANnel<x>:INSTrument:BWIDth (Query only)

Queries the per channel instrument bandwidth of the measurement. The parameter <x>=1 to 8 represents the channel index.

## Conditions

Should be connected to MSO/LPD/SX to set or query.

## Group

Sense commands

## Syntax

SENSe:SIGNalvu:ACQuisition:PER:CHANnel<x>:INSTrument:BWIDth? (Query)

## Returns

<NRf>

## Examples

`SENSe:SIGNalvu:ACQuisition:PER:CHANnel1:INSTrument:BWIDth? 500.0000E+06` indicates that instrument bandwidth of channel 1 is 500 MHz.

## **[ :SENSe ]:SIGNalvu:ACQuisition:PER:CHANnel<x>:REFlevel**

Sets or queries per channel reference level of the measurement. The parameter <x>=1 to 8 represents the channel index.

### Conditions

Should be connected to MSO/LPD/SX to set or query.

### Group

Sense commands

### Syntax

```
[ :SENSe ]:SIGNalvu:ACQuisition:PER:CHANnel<x>:REFlevel<value>
```

```
[ :SENSe ]:SIGNalvu:ACQuisition:PER:CHANnel<x>:REFlevel? (Query)
```

### Argument

<value>::<NRf> specifies the reference level.

### Returns

<NRf>

## Examples

`SENSe:SIGNalvu:ACQuisition:PER:CHANnel1:REFlevel 20.0000000000`, sets the reference level of the channel 1 measurement to 20 dB.

## **[ SENSE ]:SIGNalvu:ACQuisition:PER:CHANnel<x>:TERMination**

Sets or queries the per channel termination of the measurement. The parameter <x>=1 to 8 represents the channel index.

### Conditions

Should be connected to MSO/LPD/SX to set or query.

### Group

Sense commands

### Syntax

```
SENSe:SIGNalvu:ACQuisition:PER:CHANnel<x>:TERMination {FIFTyohms|ONEMegohms}
```

```
SENSe:SIGNalvu:ACQuisition:PER:CHANnel<x>:TERMination? (Query)
```

## Argument

`FIFTYohms` sets the channel input resistance as 50  $\Omega$ .

`ONEMegaohms` sets the channel input resistance to 1 M $\Omega$ .

## Returns

<NRf>

## Examples

`SENSe:SIGNAlvu:ACQuisition:PER:CHANnel<x>:TERMination FIFTYohms`, sets the termination of the Channel 1 to 50 ohms.

## [SENSe]:SIGNAlvu:ACQuisition:RESet (No query form)

Clears the acquisition data and restarts the SignalVu acquisitions from the oscilloscope.

## Conditions

Should be connected to MSO/LPD/SX.

## Group

Sense commands

## Syntax

```
SENSe:SIGNAlvu:ACQuisition:RESet
```

## Examples

`SENSe:SIGNAlvu:ACQuisition:RESet` resets the Signalvu acquisitions from the oscilloscope.

## [ :SENSe]:SIGNAlvu:ACQuisition:SELEcted:CHANnel:TYPE

Sets or queries the selected channel type of the measurement.

## Conditions

Should be connected to SX to set or query.

## Group

Sense commands

## Syntax

```
[ :SENSe]:SIGNAlvu:ACQuisition:SELEcted:CHANnel:TYPE? (Query)
```

```
[ :SENSe]:SIGNAlvu:ACQuisition:SELEcted:CHANnel:TYPE {ATI | NONAti}
```

## Arguments

ATI sets the channel type to ATI.

NONAti sets the channel type to NON ATI.

## Examples

`:SENSe:SIGNalvu:ACQuisition:SELEcted:CHANnel:TYPE ATI`, sets the selected channel type of measurement to ATI.

## [SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CALIbrate:IQ

Use this command to perform null IQ offset operation for each source. It is effective from the next acquisition.

The parameter <x>=1 to 8 represents the number of sources.

## Related Commands

`SENSe:SIGNalvu:ACQuisition:NULL:IQ:OFFSet:CLEAR`

## Conditions

Should be connected to MSO/LPD/SX to query.

## Group

Sense commands

## Syntax

`[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CALIbrate:IQ`

## Examples

`SENSe:SIGNalvu:ACQuisition:SOURce1:CALIbrate:IQ`

## [SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel[:RF]

Sets or queries the channel of measurement source for RF data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

`[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel[:RF] {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}`

`[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel[:RF]? (Query)`

## Arguments

The parameter <x>=1 to 8 representing Range of Source.

## Examples

`SENSe:SIGNalvu:ACQuisition:SOURce1:CHANnel:RF?` might return CH1, which indicates the channel of measurement source for RF data type.

## **[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:I**

Sets or queries the channel of measurement source for I data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:I {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:I? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of Source.

## Examples

`SENSe:SIGNalvu:ACQuisition:SOURce1:CHANnel:I?` might return CH1, which indicates the channel of measurement source for I data type.

## **[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:Q**

Sets or queries the channel of measurement source for Q data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands



## Syntax

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:Q {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:Q? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of source.

## Examples

SENSe:SIGNalvu:ACQuisition:SOURce1:CHANnel:Q? might return CH1, which indicates the channel of measurement source for Q data type.

## [SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:IPLus

Sets or queries the channel of measurement source for IPlus data type .

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:IPLus {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:IPLus? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of source.

## Examples

SENSe:SIGNalvu:ACQuisition:SOURce1:CHANnel:IPLus? might return CH1, which indicates the channel of measurement source IPlus data type.

## [SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:IMINus

Sets or queries the channel of measurement source for IMinus data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:IMINus {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:IMINus? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of source.

## Examples

SENSE:SIGNALvu:ACQuisition:SOURce1:CHANnel:IMINus? might return CH1, which indicates the channel of measurement source for IMinus data type.

## [SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:QPLus

Sets or queries the channel of measurement source for QPlus data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:QPLus {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:QPLus? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of source.

## Examples

SENSE:SIGNALvu:ACQuisition:SOURce1:CHANnel:QPLus? might return CH1, which indicates the channel of measurement source for QPlus data type.

## [SENSE]:SIGNALvu:ACQuisition[:SOURCE<x>]:CHANnel:QMINus

Sets or queries the channel of measurement source for QMinus data type.

## Conditions

Should be connected to MSO/LPD to Set or Query.

Source keyword is optional, if not present Source1 will be default source.

## Group

Sense commands

## Syntax

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:QMINus {"CH1" | "CH2" | "CH3" | "CH4" | "CH5" | "CH6" | "CH7"|"CH8"}
```

```
[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:CHANnel:QMINus? (Query)
```

## Arguments

The parameter <x>=1 to 8 representing Range of source.

## Examples

`SENSe:SIGNalvu:ACQuisition:SOURce1:CHANnel:QMINus?` might return CH1, which indicates the channel of measurement source for QMinus data type.

## [ :SENSe ]:SIGNalvu:ACQuisition[ :SOURce<x> ]: DOWNconverter:INJEctionside

Sets or queries the downconverter injection side. The parameter <x>=1 to 8 represents the range of the source.

## Conditions

Should be connected to MSO/LPD/RSA/SX to set or query.

Source keyword is optional, if not present, Source1 will be default source.

## Group

Sense commands

## Syntax

```
:SENSe:SIGNalvu:ACQuisition:SOURce<x>:DOWNconverter:INJEctionside {LOWI | HIGH }
```

```
:SENSe:SIGNalvu:ACQuisition:SOURce<x>:DOWNconverter:INJEctionside? (Query)
```

## Examples

`ACQuisition:SOURce1: DOWNconverter:INJEctionside?` Might return LOWI which indicates the downconverter injection side set to low.

## **[ :SENSe ]:SIGNAlvu:ACQuisition[ :SOURce<x> ]: DOWNconverter:INSTrument:BANDwidth:MAXFreq**

Sets or queries the maximum range of the instrument bandwidth. The parameter <x>=1 to 8 represents the range of the source.

### **Conditions**

Should be connected to MSO/LPD/RSA/SX to set or query.

Source keyword is optional, if not present, Source1 will be default source.

### **Group**

Sense commands

### **Syntax**

```
[ :SENSe ]:SIGNAlvu:ACQuisition[ :SOURce<x> ]:DOWNconverter:INSTrument:BANDwidth:MAXFreq <value>
```

```
[ :SENSe ]:SIGNAlvu:ACQuisition[ :SOURce<x> ]:DOWNconverter:INSTrument:BANDwidth:MAXFreq? (Query)
```

### **Argument**

<value>: :<NRf> specifies the minimum value of the instrument bandwidth being used.

### **Returns**

<NRf>

### **Examples**

```
SENSe:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:INSTrument:BANDwidth:MaxFreq?  
Might return 7.5000000000E+9, which indicates the maximum range of the instrument bandwidth.
```

## **[ :SENSe ]:SIGNAlvu:ACQuisition[ :SOURce<x> ]:DOWNconverter:INSTrument:BAN Dwidth:MINFreq**

Sets or queries the minimum range of instrument bandwidth. The parameter <x>=1 to 8 represents the range of the source.

### **Conditions**

Should be connected to MSO/LPD/RSA/SX to set or query.

Source keyword is optional, if not present, Source1 will be default source.

### **Group**

Sense commands

### **Syntax**

```
[ :SENSe ]:SIGNAlvu:ACQuisition[ :SOURce<x> ]:DOWNconverter:INSTrument:BANDwidth:MINFreq <value>
```

```
[ :SENSe]:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:INSTrument:BANdwidth:MINFre
q? (Query)
```

### Argument

<value>::<NRf> specifies the minimum value of the instrument bandwidth being used.

### Returns

<NRf>

### Examples

```
SENSe:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:INSTrument:BANdwidth:MINFreq?
Might return 0 Hz which indicates the minimum range of the instrument bandwidth.
```

## [ :SENSe]:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:LOFReq

Sets or queries the downconverter LO frequency. The parameter <x>=1 to 8 represents the range of the source.

### Conditions

Should be connected to MSO/LPD/RSA/SX to set or query.

Source keyword is optional, if not present, Source1 will be default source.

### Syntax

```
[ :SENSe]:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:LOFReq <value>
[:Senses]:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:LOFReq? (Query)
```

### Argument

<value>::<NRf> specifies the LO frequency.

### Returns

<NRf>

### Examples

```
:SENSe:SIGNAlvu:ACQuisition:SOURce1:DOWNconverter:LOFReq 3.000000000E+9, sets the LO
frequency to 3 GHz.
```

## [ :SENSe]:SIGNAlvu:ACQuisition[:SOURce<x>]:DOWNconverter:STATe

Sets or queries the downconverter state. The parameter <x>=1 to 8 represents the range of the source.

### Conditions

Should be connected to MSO/LPD/RSA/SX to set or query.

Source keyword is optional, if not present, Source1 will be default source.

## Group

Sense commands

## Syntax

```
[[:SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:DOWNconverter:STATe{ TRUE | FALSE}  
[:SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:DOWNconverter:STATe? (Query)
```

## Examples

`:SENSe:SIGNalvu:ACQuisition:SOURce1:DOWNconverter:STATe TRUE` enables the downconverter for Source 1.

## **[SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:PERSrc:REFlevel**

Sets or queries the Reference level Per source .

## Conditions

Should be connected to MSO/LPD to Set or Query.

## Group

Sense comands

## Syntax

```
[[:SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:PERSrc:REFlevel? (Query)  
[:SENSe]:SIGNalvu:ACQuisition[:SOURce<x>]:PERSrc:REFlevel <Value>
```

## Arguments

<value>::<NRf> specifies the global reference level

Range: {-130 to 30} in dBm

## Returns

<NRf>

## Examples

`SENSe:SIGNalvu:ACQuisition:SOURce1:PERSrc:REFlevel?` might return 5, which indicates the ref level value of source 1 is 5.

## **[SENSe]:SPECtrum[:MEASview<y>]:SELEcted:SOURce**

Sets or queries the Source of the selected Measurement view in pulse trace

## Conditions

Measurement views: Spectrum with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:SELEcted:SOURce
{"Source1"|"Source2"|"Source3"|"Source4"}
[SENSe]:SPECTrum[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

<y>=1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1

## Examples

:SENSe:SPECTrum:TRACE:MEASVIEW1:SELEcted:SOURce? Queries the Source of the Spectrum Measurement View 1.

## [SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth (RBW). Programming a specified RBW sets [SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO OFF.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution] <value>
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO
```

## Arguments

<value>::<NRf> specifies the RBW. Range: 1 Hz to 10 MHz.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:SPECTrum:MEASVIEW2:BANDwidth:RESolution 200kHz sets the RBW to 200 kHz.

## **[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:ACTual? (Query Only)**

Queries the actual resolution bandwidth (RBW) in the spectrum measurement.

### **Conditions**

Measurement views: Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:ACTual?
```

### **Arguments**

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### **Returns**

<NRf> The actual RBW in Hz.

### **Examples**

`SENSE:SPECTRUM:MEASVIEW2:BANDWIDTH:RESOLUTION:ACTUAL?` might return `299.624E+3`, indicating that the actual RBW is 299.624 kHz.

## **[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO**

Determines whether to set the resolution bandwidth (RBW) automatically or manually in the spectrum measurement.

### **Conditions**

Measurement views: Spectrum

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO?
```

### **Arguments**

OFF or 0 specifies that the resolution bandwidth is set manually using the `[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]` command.

ON or 1 specifies that the resolution bandwidth is set automatically.



The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSE:SPECTRUM:MEASVIEW2:BANDWIDTH:RESOLUTION:AUTO ON` sets the resolution bandwidth automatically.

## **[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE**

Determines whether to enable or disable the RBW process.



**Note:** *This is an obsolete command.*

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE { OFF| ON | 0 | 1 }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:MODE?
```

## Arguments

OFF or 0 disables the RBW process. You can select the FFT window using the `[SENSe]:SPECTrum[:MEASview<y>]:FFT:WINDow` command.

ON or 1 enables the RBW process. Refer to the `[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]` command to set the RBW.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSE:SPECTRUM:MEASVIEW2:BANDWIDTH:RESOLUTION:MODE ON` enables the RBW process.

## **[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo**

Sets or queries the video bandwidth (VBW). Programming a specified VBW sets `[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe OFF`.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo <value>  
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo?
```

## Arguments

<value>::<Nrf> specifies the VBW.

Range: Current RBW/104 (1 Hz minimum) to Current RBW.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
SENSE:SPECTRUM:MEASVIEW2:BANDWIDTH:VIDEO 200kHz sets the VBW to 200 kHz.
```

## [SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the spectrum measurement.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe { OFF | ON | 0 | 1 }  
[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}:VIDeo:STATe?
```

## Arguments

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
SENSE:SPECTRUM:MEASVIEW2:BANDWIDTH:VIDEO:STATE ON enables the VBW
```

## [SENSe]:SPECTrum[:MEASview<y>]:CLEAr:RESults (No Query Form)

Restarts multi-trace functions (Average and Max/Min Hold).

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:CLEar:RESults
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:SPECTrum:MEASview2:CLEar:RESults restarts multi-trace functions.

## [SENSe]:SPECTrum[:MEASview<y>]:FFT:WINDow

Sets or queries the FFT window in the spectrum measurement. This command is equivalent to

```
[SENSe]:SPECTrum[:MEASview<y>]:FILTer[:SHAPE].
```

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FFT:WINDow { KAISer | MIL6db | CISPr | BH4B | UNIFORM  
| FLATtop | HANNing }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FFT:WINDow?
```

## Arguments

KAISer selects the Kaiser (RBW) window.

MIL6db selects the -6 dB RBW (MIL) window.

CISPr selects the CISPR window.

BH4B selects the Blackman-Harris 4B type window.

UNIFORM selects the uniform window.

FLATtop selects the flat-top window.

HANNing selects the Hanning window.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSE:SPECTRUM:MEASVIEW2:FFT:WINDOW HANNing` selects the Hanning window.

## [SENSE]:SPECTrum[:MEASview<y>]:FILTer[:SHAPE]

Sets or queries the filter shape in the spectrum measurement. This command is equivalent to

`[SENSE]:SPECTrum[:MEASview<y>]:FFT:WINDow.`

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

`[SENSE]:SPECTrum[:MEASview<y>]:FILTer[:SHAPE] { KAISer | MIL6db | CISPr | BH4B | UNIFORM | FLATtop | HANNing }`

`[SENSE]:SPECTrum[:MEASview<y>]:FILTer[:SHAPE]?`

## Arguments

`KAISer` selects the Kaiser (RBW) window.

`MIL6db` selects the -6 dB RBW (MIL) window.

`CISPr` selects the CISPR window.

`BH4B` selects the Blackman-Harris 4B type window.

`UNIFORM` selects the uniform window.

`FLATtop` selects the flat-top window.

`HANNing` selects the Hanning window.

## Examples

`SENSE:SPECTRUM:MEASVIEW1:FILTER:SHAPE HANNing` selects the Hanning window.

## [SENSE]:SPECTrum[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the spectrum measurement.



**Note:** The center, start and stop frequencies are set interlocking each other with the following relationships:  $(\text{start frequency}) = (\text{center frequency}) - (\text{span})/2$  and  $(\text{stop frequency}) = (\text{center frequency}) + (\text{span})/2$ .

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer <value>
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START,
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP
```

## Arguments

<value>::<NRf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
SENSE:SPECTRUM:MEASVIEW2:FREQUENCY:CENTER 7.5GHz sets the center frequency to 7.5 GHz..
```

## [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN

Sets or queries the frequency span in the spectrum measurement.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN <value>
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN?
```

## Arguments

<value>::<NRf> specifies the frequency span.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:FREQuency:SPAN 20MHz` sets the span to 20 MHz

## **[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio**

Sets or queries the ratio of span to RBW (Resolution Bandwidth) in the spectrum measurement. This command is valid when `[SENSe]:SPECTrum[:MEASview<y>]:{BANDwidth|BWIDth}[:RESolution]:AUTO` is set to On.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio <value>`

`[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:SPAN:BANDwidth[:RESolution]:RATio?`

## Arguments

`<value>::<Nrf>` specifies the ratio of span to RBW. Range: 20 to 1000. Programming a specified ratio sets the RBW (= span/ratio), which is rounded down to the nearest valid value.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:FREQuency:SPAN:BANDwidth:RESolution:RATio 200` sets the ratio to 200, setting the RBW to 200 kHz for the span of 40 MHz.

## **[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START**

Sets or queries the measurement start frequency (left edge on the graph) in the spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer` command.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START <value>
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP
```

## Arguments

<value>: :<NRf> is the measurement start frequency.

Range: (center frequency)  $\pm$ (span)/2.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:SPECTrum:MEASview2:FREQuency:START 6.95GHz sets the start frequency to 6.95 GHz.

## [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP

Sets or queries the frequency step size (the amount per press by which the up or down key changes the setting value). Programming a specified step size sets [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO OFF.

The center, start and stop frequencies are set interlocking each other. Refer to the [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer command.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP <value>
```

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO
```

## Arguments

<value>: :<NRf> specifies the frequency step size.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz

- RSA5126B – 1 to 26.5 GHz

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:FREQuency:STEP 1.5kHz` sets the step size to 1.5 kHz.

## [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO

Determines whether to set the frequency step size automatically or manually.

## Conditions

Measurement views: Spectrum

## Group

Sense commands

## Syntax

`[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO { OFF | ON | 0 | 1 }`

`[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP:AUTO?`

## Arguments

OFF or 0 specifies that the frequency step size is set manually using the `[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STEP` command.

ON or 1 specifies that the frequency step size is set automatically.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:FREQuency:STEP:AUTO ON` sets the frequency step size automatically

## [SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP

Sets or queries the measurement stop frequency (right edge on the graph) in the spectrum measurement.

The center, start and stop frequencies are set interlocking each other. Refer to the `[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:CENTer` command.

## Conditions

Measurement views: Spectrum

## Group

Sense commands



## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP <value>
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:STOP?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:FREQuency:START
```

## Arguments

<value>: :<Nrf> is the measurement start frequency.

Range: (center frequency)  $\pm$  (span)/2.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
SENSe:SPECTrum:MEASview2:FREQuency:STOp 7.05 sets the stop frequency to 7.05 GHz.
```

## [SENSe]:SPECTrum[:MEASview<y>]:LENGth

Sets or queries the spectrum length when [SENSe]:SPECTrum:TIME:MODE is INDEpendent. Programming a specified length sets [SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO OFF.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:LENGth <value>
[SENSe]:SPECTrum[:MEASview<y>]:LENGth?
```

## Related Commands

```
[SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO, [SENSe]:SPECTrum:START
```

## Arguments

<value>: :<Nrf> specifies the spectrum length.

Range: 0 to [(acquisition length) - (spectrum start)].

If [(spectrum start) + (spectrum length)] > (acquisition length), the actual spectrum length is reduced to the acquisition length.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:LENGth 38.5us` sets the spectrum length to 38.5  $\mu$ s.

## [SENSe]:SPECTrum[:MEASview<y>]:LENGth:ACTual (Query Only)

Queries the actual spectrum length.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

`[SENSe]:SPECTrum[:MEASview<y>]:LENGth:ACTual?`

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Returns

<NRf> Actual spectrum length in seconds.

## Examples

`SENSe:SPECTrum:MEASview2:LENGth:ACTual?` might return 6.337E-6, indicating that the actual spectrum length is 6.337  $\mu$ s

## [SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO

Sets or queries the spectrum length mode (automatic or manual) when `[SENSe]:SPECTrum:TIME:MODE` is INDEPENDENT.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

`[SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO { OFF | ON | 0 | 1 }`

`[SENSe]:SPECTrum[:MEASview<y>]:LENGth:AUTO?`

### Arguments

OFF or 0 sets the spectrum length manually, using the `[SENSe]:SPECTrum:LENGth` command.

ON or 1 sets the spectrum length automatically.

## Examples

`SENSe:SPECTRUM:MEASview1:LENGTH:AUTO ON` sets the spectrum length automatically.

## [SENSe]:SPECTrum[:MEASview<y>]:MAX:SPAN (No Query Form)

Sets the frequency span to the maximum full span instead of the maximum real-time span. The maximum span depends on the model.

### Conditions

Measurement views: Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:MAX:SPAN
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`SENSe:SPECTrum:MEASview2:MAX:SPAN` sets the frequency span to the maximum span.

## [SENSe]:SPECTrum[:MEASview<y>]:POINTS:COUNT

Sets or queries the number of sample points on the signal spectrum.

### Conditions

Measurement views: Spectrum

### Group

Sense commands

### Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:POINTS:COUNT { P801 | P1601 | P2401 | P3201 | P4001 | P6401 | P8001 | P10401 }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:POINTS:COUNT?
```

### Arguments

P801 sets the number of sample points to 801.

P1601 sets the number of sample points to 1601.

P2401 sets the number of sample points to 2401.

P3201 sets the number of sample points to 3201.

P4001 sets the number of sample points to 4001.

P6401 sets the number of sample points to 6401.

P8001 sets the number of sample points to 8001.

P10401 sets the number of sample points to 10401.

The parameter  $\langle y \rangle = 1$  to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default  $y$  is 1

## Examples

SENSe:SPECTrum:MEASview2:POINTs:COUNT P801 sets the number of sample points to 801

## [SENSe]:SPECTrum[:MEASview<y>]:START

Sets or queries the spectrum offset time when [SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE is INDependent.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:SPECTrum[:MEASview<y>]:START <value>

[SENSe]:SPECTrum[:MEASview<y>]:START?

## Related Commands

[SENSe]:SPECTrum:LENGth

## Arguments

<value>: : <NRf> specifies the spectrum offset time.

Range: 0 to the acquisition length.

If  $[(\text{spectrum start}) + (\text{spectrum length})] > (\text{acquisition length})$ , the actual spectrum length is reduced to the acquisition length.

The parameter  $\langle y \rangle = 1$  to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default  $y$  is 1

## Examples

SENSe:SPECTRUM:MEASview1:START 23.5us sets the analysis offset to 23.5  $\mu\text{s}$ .

## [SENSe]:SPECTrum[:MEASview<y>]:START:AUtO

Sets or queries the spectrum offset mode (automatic or manual) when [SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE is INDePendent.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:START:AUtO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:START:AUtO?
```

### Arguments

OFF or 0 sets the spectrum offset manually.

ON or 1 sets the spectrum length automatically.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

SENSe:SPECTRUM:MEASview1:START:AUtO ON sets the spectrum offset automatically.

## [SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE

Sets or queries the spectrum time parameter mode (automatic or manual).

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE { INDePendent | COMMon }
```

```
[SENSe]:SPECTrum[:MEASview<y>]:TIME:MODE?
```

### Related Commands

[SENSe]:ANALySis commands

### Arguments

INDePendent sets the spectrum time parameters manually, using the [SENSe]:SPECTrum:START and [SENSe]:SPECTrum:LENGth commands.

COMMon sets the spectrum time parameters automatically using the analysis time settings.

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

SENSe:SPECTRUM:MEASview1:TIME:MODE COMMon sets the spectrum time parameters automatically.

## [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}

Sets or queries the channel bandwidth for the carrier as power reference. This command is valid when [SENSe]:SPURious:REFeRence is set to CARRier.

### Conditions

Measurement views: Spurious

### Group

Sense commands

### Syntax

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth} <value>

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}?

### Arguments

<value>::<NRf> specifies the channel bandwidth.

Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

## Examples

SENSe:SPURIOUS:CARRIER:BANDWIDTH 20MHz sets the channel bandwidth to 20 MHz.

## [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration

Sets or queries the integration bandwidth to calculate the carrier power. This command is valid when [SENSe]:SPURious:REFeRence is set to CARRier.

### Conditions

Measurement views: Spurious

### Group

Sense commands

### Syntax

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration <value>

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}:INTegration?

## Arguments

<value>::<Nrf> specifies the integration bandwidth.

Range: 100 Hz to 40 MHz (Standard) / 110 MHz (Option 110).

## Examples

SENSe:SPURious:CARRier:BANDwidth:INTEGRATION 2MHz sets the integration bandwidth to 2 MHz.

## [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[ :RESolution]

Sets or queries the resolution bandwidth (RBW) to measure the carrier power. This command is valid when [SENSe]:SPURious:REFeRence is set to CARRier. Programming a specified RBW sets [SENSe]:SPURious:CARRier:BANDwidth [RESolution]:AUTO OFF.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[ :RESolution] <value>

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[ :RESolution]?

## Related Commands

[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[ :RESolution]:AUTO

## Arguments

<value>::<Nrf> specifies the RBW. Range: 1 Hz to 10 MHz.

## Examples

SENSe:SPURious:CARRier:BANDwidth:RESOLUTION 200kHz sets the RBW to 200 kHz.

## [SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[ :RESolution]:AUTO

Determines whether to set the resolution bandwidth (RBW) automatically or manually when the power reference is set to carrier ([SENSe]:SPURious:REFeRence is set to CARRier) in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]: AUTO { OFF | ON | 0 | 1 }  
[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]: AUTO?
```

## Arguments

OFF or 0 specifies that the resolution bandwidth is set manually using the `[SENSe]:SPURious:CARRier:{BANDwidth|BWIDth}[:RESolution]` command.

ON or 1 specifies that the resolution bandwidth is set automatically.

## Examples

`SENSe:SPURIOUS:CARRIER:BANDWIDTH:RESOLUTION:AUTO ON` sets the resolution bandwidth automatically.

## [SENSe]:SPURious:CARRier:DETection

Sets or queries the carrier detection method. This command is valid when `[SENSe]:SPURious:REFeRence` is set to `CARRier`.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:CARRier:DETection { AVERage | PEAK | AVGLog}  
[SENSe]:SPURious:CARRier:DETection?
```

## Arguments

`AVERage` selects the average detection.

`PEAK` selects the peak detection.

`AVGLog` selects the average (of logs) detection.

## Examples

`SENSe:SPURIOUS:CARRIER:DETECTION PEAK` selects the peak detection.

## [SENSe]:SPURious:CARRier:FREQuency

Sets or queries the carrier frequency in the Spurious measurement. This command is valid when `[SENSe]:SPURious:REFeRence` is set to `CARRier`.

## Conditions

Measurement views: Spurious



## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:CARRier:FREQuency <value>
```

```
[SENSe]:SPURious:CARRier:FREQuency?
```

## Arguments

<value>: : <NRf> specifies the carrier frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

SENSe:SPURIOUS:CARRIER:FREQUENCY 7.5GHz sets the carrier frequency to 7.5 GHz.

## [SENSe]:SPURious:CARRier:THReshold

Sets or queries the threshold level to detect the carrier in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFeRence is set to CARRier.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:CARRier:THReshold <value>
```

```
[SENSe]:SPURious:CARRier:THReshold?
```

## Arguments

<value>: : <NRf> specifies the threshold level above which the input signal is determined to be a carrier. Range: -170 to +50 dBm. The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

## Examples

SENSe:SPURIOUS:CARRIER:THRESHOLD -25 sets the carrier threshold level to -25 dBm.

## [SENSe]:SPURious:CLEar:RESults (No Query Form)

Restarts multi-trace functions (Average and Max Hold).

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:CLEar:RESults
```

## Arguments

None

## Examples

`SENSe:SPURIOUS:CLEAR:RESULTS` restarts multi-trace functions.

## [SENSe]:SPURious[:FREQuency]:OVERlap? (Query Only)

Queries whether any of the frequency ranges (A to T) overlap, including the carrier when `[SENSe]:SPURious:REFeRence` is set to `CARRier`.



**Note:** *If there are any overlaps between the ranges, the measurement will not run correctly.*

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious[:FREQuency]:OVERlap?
```

## Arguments

None

## Returns

0 (no overlap) or 1 (overlap).

## Examples

`SENSe:SPURIOUS:FREQUENCY:OVERLAP?` might return 1, indicating that some frequency ranges overlap.

## [SENSe]:SPURious:LIST

Sets or queries how to list the spurious signals in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:LIST { ALL | OVERlimit }
```

```
[SENSe]:SPURious:LIST?
```

## Arguments

ALL lists all of the detected spurious signals.

OVERlimit lists the spurious signals exceeding the limits. Use the [SENSe]:SPURious:RANGe<x>:LIMit command group to set the limits.

## Examples

SENSe:SPURIOUS:LIST ALL lists all of the detected spurious signals.

## [SENSe]:SPURious:MODE

Sets or queries the frequency range mode in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:MODE { MULTi | SINGLe }
```

```
[SENSe]:SPURious:MODE?
```

## Arguments

MULTi displays all of the ranges that are enabled. Use the [SENSe]:SPURious: RANGe<x>:STATe command to enable the range.

SINGLe displays only the range that the selected spurious signal is in. Use the DISPLAY:SPURious:SElect:NUMBER command to select the spurious signal.

## Examples

SENSe:SPURIOUS:MODE MULTi displays all of the enabled ranges.

## [SENSe]:SPURious:OPTimization

Sets or queries the method of optimizing the gain and input bandwidth in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Sense commands

### Syntax

```
[SENSe]:SPURious:OPTimization { AUTO | MINTime | MAXDynrange | MINNoise }  
[SENSe]:SPURious:OPTimization?
```

### Arguments

`AUTO` optimizes automatically the gain and input bandwidth.

`MINTime` optimizes the gain and input bandwidth to minimize sweep time.

`MAXDynrange` optimizes the gain and input bandwidth to maximize the dynamic range.

`MINNoise` optimizes the gain and input bandwidth to minimize noise.

### Examples

`SENSe:SPURIOUS:OPTIMIZATION AUTO` optimizes automatically the gain and input bandwidth.

## [SENSe]:SPURious:POINTs:COUNT

Sets or queries the number of sample points on the spectrum trace per range in the Spurious measurement.

### Conditions

Measurement views: Spurious

### Group

Sense commands

### Syntax

```
[SENSe]:SPURious:POINTs:COUNT { P801 | P2401 | P4001 | P8001 | P10401 }  
[SENSe]:SPURious:POINTs:COUNT?
```

### Arguments

`P801` sets the trace points to 801 per range.

`P2401` sets the trace points to 2401 per range.

`P4001` sets the trace points to 4001 per range.

`P8001` sets the trace points to 8001 per range.

P10401 sets the trace points to 10401 per range.

## Examples

`SENSe:SPURIOUS:POINTS:COUNT P801` sets the trace points to 801 per range.

## [SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo

Sets or queries the video bandwidth (VBW) in the specified frequency range. Programming a specified VBW sets

`[SENSe]:SPURious:BANDwidth:VIDeo STATE OFF`.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

`[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo <value>`

`[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo?`

## Related Commands

`[SENSe]:SPECTrum:{BANDwidth|BWIDth}:VIDeo:STATE`

## Arguments

<value>: : <Nrf> specifies the VBW.

Range: Current RBW/104 (1 Hz minimum) to Current RBW.

## Examples

`SENSe:SPURIOUS:RANGE1:BANDWIDTH:VIDEO 200kHz` sets the VBW to 200 kHz for Range A.

## [SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe

Determines whether to enable or disable the video bandwidth (VBW) in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe { OFF | ON| 0 | 1 }  
[SENSe]:SPURious:RANGe<x>:BANDwidth:VIDeo:STATe?
```

## Arguments

OFF or 0 disables the VBW.

ON or 1 enables the VBW.

## Examples

```
SENSe:SPURIOUS:RANGE1:BANDWIDTH:VIDEO:STATE ON enables the VBW for Range A.
```

## [SENSe]:SPURious:RANGe<x>:DETection

Sets or queries the spurious detection method in the specified frequency range. The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:DETection { AVERage | PEAK | QUASipeak | CAVERage | CPEak  
| AVGLog }  
[SENSe]:SPURious:RANGe<x>:DETection?
```

## Arguments

AVERage selects the average detection.

PEAK selects the peak detection.

QUASipeak selects the quasi-peak detection.

CAVERage selects the CISPR average detection.

CPEak selects the CISPR peak detection.

AVGLog selects the average (of logs) detection.

## Examples

```
SENSe:SPURIOUS:RANGE1:DETECTION PEAK selects the peak detection for Range A.
```

## [SENSe]:SPURious:RANGe<x>:EXCursion

Sets or queries the excursion level (how far down the signal must drop between spurious emissions) in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:EXCURsion <value>
```

```
[SENSe]:SPURious:RANGe<x>:EXCURsion?
```

## Arguments

<value>::<NRF> specifies the excursion level. A signal with amplitude less than the excursion level is considered to be a noise.  
Range: 1 to 50 dB.

## Examples

SENSe:SPURIOUS:RANGE1:EXCURSION 8 sets the excursion level to 8 dB.

## [SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]

Sets or queries the filter shape to search the specified frequency range for spurious signals.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE] { RBW | MIL6db | CISPr }
```

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]?
```

## Arguments

RBW selects the RBW filter.

MIL6db selects the -6 dB RBW (MIL) filter.

CISPr selects the CISPR filter.

## Examples

SENSe:SPURIOUS:RANGE1:FILTER:SHAPE CISPr uses the CISPR filter in Range A.

## **[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth**

Sets or queries the filter bandwidth to search the specified frequency range for spurious signals.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

### **Conditions**

Measurement views: Spurious

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth <value>
```

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth?
```

### **Arguments**

<value>::<NRF> specifies the filter bandwidth.

The setting range varies depending on the center frequency and trace detector when using the CISPR filters and detectors. Otherwise, the bandwidth minimum is a function of the span (= (stop frequency) - (start frequency)) of the range and the maximum is fixed at 5 MHz.

By default, the value is set automatically ([SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO is ON). When the detection is set to CISPR QPk ([SENSe]:SPURious:RANGe<x>:DETEction is QUASipeak), only one value is allowed, which is set automatically.

### **Examples**

SENSe:SPURIOUS:RANGE1:FILTER:SHAPE:BANDWIDTH 200kHz sets the filter bandwidth to 200 kHz for Range A.

## **[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO**

Determines whether to set the filter bandwidth automatically or manually for the specified frequency range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

### **Conditions**

Measurement views: Spurious

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth:AUTO?
```



## Arguments

OFF or 0 specifies that the filter bandwidth is set manually using the `[SENSe]:SPURious:RANGe<x>:FILTer[:SHAPE]:BANDwidth` command.

ON or 1 specifies that the filter bandwidth is set automatically.

## Examples

`SENSe:SPURIOUS:RANGE1:FILTER:SHAPE:BANDWIDTH:AUTO ON` sets the filter bandwidth automatically for Range A.

## [SENSe]:SPURious:RANGe<x>:FREQuency:START

Sets or queries the start frequency of the specified range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.



**Note:** *The frequency ranges must not be overlapped. Use the `[SENSe]:SPURious[:FREQuency]:OVERlap?` query to check whether there is any overlap.*

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

`[SENSe]:SPURious:RANGe<x>:FREQuency:START <value>`

`[SENSe]:SPURious:RANGe<x>:FREQuency:START?`

## Arguments

`<value>::<NRf>` specifies the start frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Examples

`SENSe:SPURIOUS:RANGE1:FREQUENCY:START 1.615GHz` sets the start frequency of Range A to 1.615 GHz.

## [SENSe]:SPURious:RANGe<x>:FREQuency:STOP

Sets or queries the stop frequency of the specified range in the Spurious measurement.

The parameter <x> = 1 to 20, representing Range A to T, respectively.



**Note:** *The frequency ranges must not be overlapped. Use the `[SENSe]:SPURious[:FREQuency]:OVERlap?` query to check whether there is any overlap.*

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:FREQUency:STOP <value>
```

```
[SENSe]:SPURious:RANGe<x>:FREQUency:STOP?
```

## Arguments

<value>::<Nrf> specifies the stop frequency.

Range:

- RSA5103B – 0 to 3 GHz
- RSA5106B – 0 to 6.2 GHz
- RSA5115B – 0 to 15 GHz
- RSA5126B – 0 to 26.5 GHz

## Examples

`SENSe:SPURIOUS:RANGE1:FREQUENCY:STOP 1.715GHz` sets the stop frequency of Range A to 1.715 GHz.

## [SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START

Sets or queries the absolute amplitude of the limits at the start (left edge) of the specified range in the Spurious measurement. This command is valid when `[SENSe]:SPURious:RANGe<x>:LIMit:MASK` is set to ABS, AND, or OR.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START <value>
```

```
[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START?
```

## Arguments

<value>::<Nrf> specifies the absolute start amplitude of the limits.

Range: -170 to +50 dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` or `UNIT:POWer` command.

## Examples

`SENSe:SPURIOUS:RANGE1:LIMIT:ABSOLUTE:START -30` sets the absolute start amplitude of the limits for Range A to -30 dBm.

## [SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP

Sets or queries the absolute amplitude of the limits at the stop (right edge) of the specified range in the Spurious measurement. This command is valid when `[SENSe]:SPURious:RANGe<x>:LIMit:MASK` is set to ABS, AND, or OR.

The parameter `<x>` = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

`[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP <value>`

`[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP?`

## Related Commands

`[SENSe]:POWer:UNITs`, `UNIT:POWer`

## Arguments

`<value>::<NRf>` specifies the absolute stop amplitude of the limits.

Range: -170 to +50 dBm.

The unit can be changed by the `[SENSe]:POWer:UNITs` or `UNIT:POWer` command.

## Examples

`SENSe:SPURIOUS:RANGE1:LIMIT:ABSOLUTE:STOP -10` sets the absolute stop amplitude of the limits for Range A to -10 dBm.

## [SENSe]:SPURious:RANGe<x>:LIMit:MASK

Sets or queries the limit mask function mode for the specified range in the Spurious measurement.

The parameter `<x>` = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:MASK{ABS|REL|AND|OR|OFF}
```

```
[SENSe]:SPURious:RANGe<x>:LIMit:MASK?
```

## Arguments

The following table lists the arguments.

**Table 54: Limit mask mode**

Argument	Description
ABS	Failure is detected when one of the spurious signals is larger than the absolute amplitude limit.
REL	Failure is detected when one of the spurious signals is larger than the relative amplitude limit.
AND	Failure is detected when one of the spurious signals is larger than the absolute AND relative amplitude limits.
OR	Failure is detected when one of the spurious signals is larger than the absolute OR relative amplitude limit.
OFF	Disables the mask.

To set the absolute amplitude limits, use the `[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:START` and `[SENSe]:SPURious:RANGe<x>:LIMit:ABSolute:STOP` commands.

To set the relative amplitude limits, use the `[SENSe]:SPURious:RANGe<x>:LIMit:RELative:START` and `[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP` commands.

## Examples

`SENSe:SPURIOUS:RANGE1:LIMIT:MASK ABS` specifies that failure is detected when one of the spurious signals is larger than the absolute amplitude limit in Range A.

## [SENSe]:SPURious:RANGe<x>:LIMit:RELative:START

Sets or queries the relative amplitude of the limits at the start (left edge) of the specified range in the Spurious measurement. This command is valid when `[SENSe]:SPURious:RANGe<x>:LIMit:MASK` is set to REL, AND, or OR.

The parameter `<x>` = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:START <value>
```

```
[SENSe]:SPURious:RANGe<x>:LIMit:RELative:START?
```

## Arguments

`<value>::<NRf>` specifies the relative start amplitude of the limits.

Range: -100 to 0 dB.

Use the `[SENSe]:SPURious:REFeRence` command to select the power reference.

## Examples

`SENSe:SPURIOUS:RANGE1:LIMIT:RELATIVE:START -30` sets the relative start amplitude of the limits for Range A to -30 dB.

## [SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP

Sets or queries the relative amplitude of the limits at the stop (right edge) of the specified range in the Spurious measurement. This command is valid when `[SENSe]:SPURious:RANGe<x>:LIMit:MASK` is set to REL, AND, or OR.

The parameter `<x>` = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

`[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP <value>`

`[SENSe]:SPURious:RANGe<x>:LIMit:RELative:STOP?`

## Arguments

`<value>::<Nrf>` specifies the relative stop amplitude of the limits.

Range: -100 to 0 dB.

Use the `[SENSe]:SPURious:REFeRence` command to select the power reference.

## Examples

`SENSe:SPURIOUS:RANGE1:LIMIT:RELATIVE:STOP -10` sets the relative stop amplitude of the limits for Range A to -10 dB.

## [SENSe]:SPURious:RANGe<x>:STATe

Determines whether to enable or disable the frequency range in the Spurious measurement.

The parameter `<x>` = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:STATe { OFF | ON | 0 | 1 }  
[SENSe]:SPURious:RANGe<x>:STATe?
```

## Arguments

OFF or 0 disables the frequency range.

ON or 1 enables the frequency range.

## Examples

```
SENSe:SPURIOUS:RANGE1:STATE ON enables Range A.
```

## [SENSe]:SPURious:RANGe<x>:THReshold

Sets or queries the threshold level to detect spurious signals in the specified frequency range.

The parameter <x> = 1 to 20, representing Range A to T, respectively.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:RANGe<x>:THReshold <value>  
[SENSe]:SPURious:RANGe<x>:THReshold?
```

## Arguments

<value>::<Nrf> specifies the threshold level above which the signal is determined to be spurious. Range: -50 to +30 dBm.

The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

## Examples

```
SENSe:SPURIOUS:RANGE1:THRESHOLD -25 sets the threshold level to -25 dBm in Range A.
```

## [SENSe]:SPURious:REFerence

Sets or queries the power reference in the Spurious measurement.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:REFeRence { CARRier | MANual | NONE }
[SENSe]:SPURious:REFeRence?
```

## Arguments

CARRier uses the carrier as the power reference.

MANual sets the power reference using the [SENSe]:SPURious:REFeRence:MANual:POWer command.

NONE uses no reference.

## Examples

```
SENSe:SPURIOUS:REFERENCE CARRier uses the carrier as the power reference.
```

## [SENSe]:SPURious:REFeRence:MANual:POWer

Sets or queries the reference power level in the Spurious measurement. This command is valid when [SENSe]:SPURious:REFeRence is set to MANual.

## Conditions

Measurement views: Spurious

## Group

Sense commands

## Syntax

```
[SENSe]:SPURious:REFeRence:MANual:POWer <value>
[SENSe]:SPURious:REFeRence:MANual:POWer?
```

## Arguments

<value>::<NRf> specifies the reference power level. Range: -170 to +50 dBm.

The unit can be changed by the [SENSe]:POWer:UNITs or UNIT:POWer command.

## Examples

```
SENSe:SPURIOUS:REFERENCE:MANUAL:POWER -25 sets the reference power level to -25 dBm.
```

## [SENSe]:SQuality[:MEASview<y>]:EVM:ALL[:STATe]

Sets or queries whether or not the display of both EVM and Offset EVM results is enabled. Offset EVM appears only for OQPSK and SOQPSK modulation types.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Sense commands

## Syntax

```
[SENSe]:SQUality[:MEASview<y>]:EVM:ALL[:STATe]  
[SENSe]:SQUality[:MEASview<y>]:EVM:ALL[:STATe]?
```

## Arguments

OFF or 0 disables the local lockout operation.

ON or 1 enables the local lockout operation.

## [SENSe]:SQUality[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Signal quality for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal Quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:SQUality[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source  
3" | "Source 4"}  
[SENSe]:SQUality[:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

SENSe:SQUality:MEASview1:SELECTED:SOURce? queries the Source for Signal quality Measurement View 1.



## [SENSe]:STABle[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Symbol table for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Symbol table

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod symbol table).

Source cannot be changed when the acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

```
[SENSe]:STABle [:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3"
| "Source 4"}
```

```
[SENSe]:STABle [:MEASview<y>]:SELEcted:SOURce?
```

### Arguments

The parameter <y>=1 to 8 represents measurement index based on the number of sources of connected instrument, where MEASview keyword is optional by default and y is 1.

### Examples

`SENSe:STABle:MEASview1:SELEcted:SOURce?` queries the Source for Symbol table Measurement View 1.

## [SENSe]:SVPC:ACQuisition:CONTRol:TRIGger

Sets or queries the Trigger Settings checkbox state.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
[SENSe]:SVPC:ACQuisition:CONTRol:TRIGger { OFF | ON | 0 | 1 }
```

```
[SENSe]:SVPC:ACQuisition:CONTRol:TRIGger?
```

### Arguments

OFF or 0 disables the Trigger Settings.

ON or 1 enables the Trigger Settings.

## Examples

`SENSe:SVPC:ACQuisition:CONTRol:TRIGger ON` enables the Trigger setting.

## [SENSe]:SVPC:ACQuisition:CONTRol:CHANnel

Sets or queries the Channel Settings checkbox state.

### Conditions

Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:SVPC:ACQuisition:CONTRol:CHANnel { OFF | ON | 0 | 1 }  
[SENSe]:SVPC:ACQuisition:CONTRol:CHANnel?
```

### Arguments

OFF or 0 disables the channel settings.

ON or 1 enables the channel settings.

## Examples

`SENSe:SVPC:ACQuisition:CONTRol:CHANnel ON` enables the Channel Settings.

## [SENSe]:SVPC:ACQuisition:FFRame:ACTual? (Query Only)

Queries the actual number of Fast Frames.

### Conditions

Requires an RSA be connected with SignalVu-PC. Measurement views: All

### Group

Sense commands

### Syntax

```
[SENSe]:SVPC:ACQuisition:FFRame:ACTual?
```

### Arguments

None

## Returns

<NR1> The actual number of Fast Frames.

## Examples

`SENSe:SVPC:ACQUISITION:FFRAME:ACTUAL?` might return 178, indicating that the actual number of Fast Frames is 178.

## [SENSe]:SVPC:ACQuisition:FFRame:LIMit

Sets or queries the limit number of Fast Frames.

## Conditions

Requires an RSA be connected with SignalVu-PC. Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:SVPC:ACQuisition:FFRame:LIMit
```

```
[SENSe]:SVPC:ACQuisition:FFRame:LIMit?
```

## Arguments

<value>::<NR1> specifies the fast frame limit. Range: 1 to 65535 frames.

Where

1 to 65535 frames for RSA 5000A series. For others, the range is 1 to 220 -1 frames.

## Examples

`SENSe:SVPC:ACQUISITION:FFRAME:LIMIT 500` sets the Fast Frame limit to 500.

## [SENSe]:SVPC:ACQuisition:FFRame:STATe

Determines whether to enable or disable the Fast Frame.

## Conditions

Requires an RSA be connected with SignalVu-PC. Measurement views: All

## Group

Sense commands

## Syntax

```
[SENSe]:SVPC:ACQuisition:FFRame:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:SVPC:ACQuisition:FFRame:STATe?
```

## Arguments

OFF or 0 disables the Fast Frame.

ON or 1 enables the Fast Frame.

## Examples

`SENSe:SVPC:ACQUISITION:FFRAME:STATE ON` enables the Fast Frame.

## [SENSe]:TDIagram[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Source of Trellis diagram for a selected Measurement view number.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Trellis diagram).

Source cannot be changed when the acquisition or analysis is in progress. Stop the acquisition and change the Source.

## Group

Sense commands

## Syntax

```
[SENSe]:TDIagram [:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:TDIagram [:MEASview<y>]:SELEcted:SOURce?
```

## Arguments

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1.

## Examples

`SENSe:TDIagram:MEASview1:SELEcted:SOURce?` queries the Source for Trellis diagram Measurement View 1.

## SENSe:TOFF:POWER:SPECtral:DENSity

Sets or queries the limit set for comparison of the Toff power spectral density value in the Limits tab of the LTE Power vs Time display settings control panel.

## Conditions

Measurement view: LTE Power vs Time

## Group

Sense commands

## Syntax

```
SENSe:TOFF:POWEr:SPECTral:DENSity <value>
```

```
SENSe:TOFF:POWEr:SPECTral:DENSity?
```

## Arguments

<value>::<Nrf> specifies the Toff power spectral density value.

## Returns

<Nrf>

## Examples

SENSe:TOFF:POWEr:SPECTral:DENSity might return  $-83.000000000$ , indicating that the Toff power spectral density is set to  $-83$  dBm.

## [SENSe]:TOVerview[:MEASview<y>]:FREQuency:CENTer

Sets or queries the center frequency in the time overview.

## Conditions

Measurement views: Time overview

## Group

Sense commands

## Syntax

```
[SENSe]:TOVerview[:MEASview<y>]:FREQuency:CENTer <value>
```

```
[SENSe]:TOVerview[:MEASview<y>]:FREQuency:CENTer?
```

## Arguments

<value>::<Nrf> specifies the center frequency.

Range:

- RSA5103B – 1 to 3 GHz
- RSA5106B – 1 to 6.2 GHz
- RSA5115B – 1 to 15 GHz
- RSA5126B – 1 to 26.5 GHz

## Returns

<Nrf>

## Examples

`SENSe:TOVERVIEW:MEASVIEW1:FREQUENCY:CENTER 7.5GHz` sets the center frequency to 7.5 GHz.

## **[SENSe]:TOVerview[:MEASview<y>]:MAXTracepoints**

Sets or queries the maximum trace points in the time overview.

### Conditions

Measurement views: Time overview

### Group

Sense commands

### Syntax

```
[SENSe]:TOVerview[:MEASview<y>]:MAXTracepoints { ONEK | TENK | HUNDredk |  
NEVerdecimate }
```

```
[SENSe]:TOVerview[:MEASview<y>]:MAXTracepoints?
```

### Arguments

`ONEK` sets the maximum trace points to 1 k.

`TENK` sets the maximum trace points to 10 k.

`HUNDredk` sets the maximum trace points to 100 k.

`NEVerdecimate` never decimates the trace points.

### Returns

<NRf>

## Examples

`SENSe:TOVERVIEW:MEASVIEW1:MAXTRACEPOINTS TENK` sets the maximum trace points to 10 k.

## **[SENSe]:TOVerview[:MEASview<y>]:TIME:PER:DIVision**

This command sets or queries the time per division (dB/div) value in the Time Overview display.

### Conditions

Measurement view: Time Overview

### Group

Sense commands

### Syntax

```
[SENSe]:TOVerview[:MEASview<y>]:TIME:PER:DIVision <NRf>
```

```
[SENSe]:TOVerview[:MEASview<y>]:TIME:PER:DIVision?
```

### Arguments

<NRf> is the dB/div value.

Range: .01 to 20

### Returns

<NR2> = floating point value without an exponent, which is the time per division value.

### Examples

SENSe:TOVerview:MEASVIEW1:TIME:PER:DIV might return 10.000000000, indicating that the time per division value is set to 10.0 dB.

## [SENSe]:TOVerview[:MEASview<y>]:SELEcted:SOURce

Sets or queries the Measurement view number for the Time Overview Display

### Conditions

Measurement view: Time overview with specified Measurement view number

Source cannot be changed when acquisition or analysis is in progress. Stop the acquisition and change the Source.

### Group

Sense commands

### Syntax

```
[SENSe]:TOVerview[:MEASview<y>]:SELEcted:SOURce {"Source 1" | "Source 2" | "Source 3" | "Source 4"}
```

```
[SENSe]:TOVerview[:MEASview<y>]:SELEcted:SOURce?
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Returns

<NRf>

### Examples

SENSe:TOVERVIEW:MEASVIEW1:SELECTED:SOURCE? gives the Measurement view number for the Time overview source.

## [SENSe]:TRACe:SPURious:COUNt:RESet (No Query Form)

Clears the multi-function (Max Hold or Average) data and counter and restarts the process in the Spurious view.

## Conditions

Measurement view: Spurious Display

## Group

Sense commands

## Syntax

```
[SENSe]:TRACe:SPURious:COUNT:RESet
```

## Arguments

None

## Returns

None

## Examples

`SENSe:TRACE:SPURIOUS:COUNT:RESET` clears the multi-function data and counter and restarts the process.

## [SENSe]:TRACe<x>:SPURious:COUNT

Sets and queries the count value for the selected trace in the Spurious measurement.

## Conditions

Measurement view: Spurious Display

## Group

Sense commands

## Syntax

```
[SENSe]:TRACe<x>:SPURious:COUNT <value>
```

```
[SENSe]:TRACe<x>:SPURious:COUNT
```

## Arguments

<x> = 1,2,3,4 is the Trace

<value> = < NR1> Range is 1 to 10000

## Returns

<NR1>

## Examples

`SENSe:TRACe4:SPURious:COUNT 60` sets the count value to 60 for Trace 4.



## [SENSe]:TRACe<x>:SPURious:COUNT:ENABLE

Determines whether to enable or disable the count check box for the specified trace in the Spurious measurement.

### Conditions

Measurement view: Spurious Display

### Group

Sense commands

### Syntax

```
[SENSe]:TRACe<x>:SPURious:COUNT:ENABLE {0 | OFF | 1 | ON}
```

```
[SENSe]:TRACe<x>:SPURious:COUNT:ENABLE?
```

### Arguments

<x> = 1,2,3,4 is the Trace

ON or 1 to enable the count check box OFF or 0 to disable the count check box

### Returns

<Boolean> 0, 1

### Examples

SENSe:TRACe2:SPURious:COUNT:ENABLE ON enables the count check box for Trace 2.

## [SENSe]:TRACe<x>:SPURious:ENABLE

Determines whether to enable or disable show check box for the specified trace in the Spurious measurement.

### Conditions

Measurement view: Spurious Display

### Group

Sense commands

### Syntax

```
[SENSe]:TRACe<x>:SPURious:ENABLE {0 | OFF | 1 | ON}
```

```
[SENSe]:TRACe<x>:SPURious:ENABLE?
```

### Arguments

<x> = 1,2,3,4 is the Trace

ON or 1 to enable the show check box OFF or 0 to disable the show check box

## Returns

<Boolean> 0, 1

## Examples

`SENSe:TRACe2:SPURious:ENABle 1` enables the show check box for Trace 2.

## [SENSe]:TRACe<x>:SPURious:FREeze

Determines whether to enable or disable the freeze check box for the specified trace in the Spurious measurement.

## Conditions

Measurement view: Spurious Display

## Group

Sense commands

## Syntax

```
[SENSe]:TRACe<x>:SPURious:FREeze {0 | OFF | 1 | ON}
```

```
[SENSe]:TRACe<x>:SPURious:FREeze?
```

## Arguments

<x> = 1,2,3,4 is the Trace

ON or 1 to enable the freeze check box OFF or 0 to disable the freeze check box

## Returns

<Boolean> 0, 1

## Examples

`SENSe:TRACe3:SPURious:FREeze ON` enables the freeze check box for Trace 3.

## [SENSe]:TRACe<x>:SPURious:FUNction

Sets and queries the required function for the selected trace in Spurious measurement.

## Conditions

Measurement view: Spurious Measurement Display

## Group

Sense commands

## Syntax

```
[SENSe]:TRACe<x>:SPURious:FUNction {NONE | MAXH | AVER | AVGL}
```

```
[SENSe]:TRACe<x>:SPURious:FUNction?
```

## Arguments

<x> = 1,2,3,4 is the Trace

"NONE" to select Normal

"MAXH" to select Max Hold

"AVER" to select Avg (VRMS)

"AVGL" to select Avg (of logs)

## Returns

<String> NONE or MAXH or AVER or AVGL

## Examples

`SENSe:TRACe1:SPURious:FUNCTion NONE` sets the function to Normal for Trace 1.

## [SENSe]:TRACe<x>:SPURious:SElect

Sets the specified trace as the active trace in the Spurious display. The query returns the active trace.

## Conditions

Measurement view: Spurious Display

## Group

Sense commands

## Syntax

`[SENSe]:TRACe<x>:SPURious:SElect`

`[SENSe]:TRACe<x>:SPURious:SElect`

## Arguments

<x> = 1,2,3,4 is the Trace

## Returns

<Boolean> 0, 1

## Examples

`SENSe:TRACe2:SPURious:SElect` sets Trace 2 as the active trace.

## SENSe:TXGain:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the resolution bandwidth.

## Conditions

Measurement view: Transmission Gain

## Group

Sense commands

## Syntax

```
SENSe:TXGain:{BANDwidth|BWIDth}[[:RESolution] <value>
```

```
SENSe:TXGain:{BANDwidth|BWIDth}[[:RESolution]?
```

## Arguments

<value> ::= <NRf> specifies the RBW. Range: 1 kHz to 1 MHz.

## Returns

See Arguments.

## Examples

```
SENSe:TXGain:BANDwidth:RESolution 200E3 sets the RBW to 200 kHz.
```

## SENSe:TXGain:FREQUENCY:CENTer

Sets or queries the frequency at the center of the selected Span.

## Conditions

Measurement view: Transmission Gain

## Group

Sense commands

## Syntax

```
SENSe:TXGain:FREQUENCY:CENTer <value>
```

```
SENSe:TXGain:FREQUENCY:CENTer?
```

## Arguments

<value> ::= <NRf> specifies the center frequency.

Range: 1 kHz to 1 MHz.

## Returns

See Arguments.

## Examples

```
SENSe:TXGAIN:FREQUENCY:CENTER 1E9 sets the center frequency to 1 GHz.
```

## SENSe:TXGain:FREQUENCY:POINts

Sets or queries the number of frequency steps in the measurement range.

## Conditions

Measurement view: Transmission Gain

## Group

Sense commands

## Syntax

```
SENSe:TXGain:FREQuency:POINTs <value>
```

```
SENSe:TXGain:FREQuency:POINTs?
```

## Arguments

<value> ::= <NR1> specifies number of frequency steps.

## Returns

See Arguments.

## Examples

SENSe:TXGAIN:FREQUENCY:POINTS 101 sets the number of frequency steps to 101.

## SENSe:TXGain:FREQuency:SPAN

Sets or queries the difference between the start and stop frequencies (span).

## Conditions

Measurement view: Transmission Gain

## Group

Sense commands

## Syntax

```
SENSe:TXGain:FREQuency:SPAN <value>
```

```
SENSe:TXGain:FREQuency:SPAN?
```

## Arguments

<value> ::= <NR1> specifies the frequency span.

## Returns

See Arguments.

## Examples

SENSe:TXGAIN:FREQUENCY:SPAN 100E6 sets the frequency span to 100 MHz.

## **SENSe:TXGain:FREQuency:START**

Sets or queries the lowest frequency in the span.

### **Conditions**

Measurement view: Transmission Gain

### **Group**

Sense commands

### **Syntax**

```
SENSe:TXGain:FREQuency:START <value>
```

```
SENSe:TXGain:FREQuency:START?
```

### **Related Commands**

```
SENSe:TXGain:FREQuency:STOP
```

### **Arguments**

<value>::=<NR1> specifies the lowest frequency in the span.

### **Returns**

See Arguments.

### **Examples**

```
SENSe:TXGAIN:FREQUENCY:START 1E9 sets the lowest frequency to 1 GHz.
```

## **SENSe:TXGain:FREQuency:STEP**

Sets or queries the frequency increment. Measurements are made at frequency points from Start Freq to Stop Freq in Step Size increments.

### **Conditions**

Measurement view: Transmission Gain

### **Group**

Sense commands

### **Syntax**

```
SENSe:TXGain:FREQuency:STEP <value>
```

```
SENSe:TXGain:FREQuency:STEP?
```

### **Arguments**

<value>::=<NRf> specifies the frequency increment.

## Returns

See Arguments.

## Examples

`SENSe:TXGAIN:FREQUENCY:STEP 1E6` sets the frequency increment to 1 MHz.

## SENSe:TXGain:FREQuency:STOP

Sets or queries the highest frequency in the span.

## Conditions

Measurement view: Transmission Gain

## Group

Sense commands

## Syntax

`SENSe:TXGain:FREQuency:STOP <value>`

`SENSe:TXGain:FREQuency:STOP?`

## Related Commands

`SENSe:TXGain:FREQuency:START`

## Arguments

`<value>::=<NRf>` specifies the highest frequency.

## Returns

See Arguments.

## Examples

`SENSe:TXGAIN:FREQUENCY:STOP 2E9` sets the highest frequency to 2 GHz.

## [SENSe]:USETtings (No Query Form)

Updates the analyzer settings. This command is useful when you need to set the analyzer including the RF attenuation before taking data acquisition. Unless this command is executed, the attenuation value is not set until acquisition is taken.

## Conditions

Measurement views: All

## Group

Sense commands

## Syntax

[SENSe]:USETtings

## Arguments

None

## Examples

SENSe:USETTINGS updates settings.

## [SENSe]:WLAN:AD:ANALysis:BLOCKwise:CORRection:STATe

Enables or disables the Enable Blockwise Correction check box in the Adv Params tab of the Settings control panel for the 802.11ad displays.

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

[SENSe]:WLAN:AD:ANALysis:BLOCKwise:CORRection:STATe { 0 | OFF | 1 | ON }

[SENSe]:WLAN:AD:ANALysis:BLOCKwise:CORRection:STATe?

## Arguments

0 or OFF disables Blockwise Correction (unchecks check box)

1 or ON enables Blockwise Correction (checks check box)

## Returns

<Boolean> 0,1

## Examples

SENSe:WLAN:AD:ANALYSIS:BLOCKWISE:CORRECTION:STATE 1 enables Blockwise Correction.

## [SENSe]:WLAN:AD:ANALysis:DC:OFFSet:CORRection:STATe

Enables or disables the Enable DC Offset Correction check box for all 802.11ad measurement views (set in the Advanced Params tab of the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views



## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:DC:OFFSet:CORRection:STATe {0 | OFF| 1 | ON }
```

```
[SENSe]:WLAN:AD:ANALysis:DC:OFFSet:CORRection:STATe ?
```

## Arguments

0 or OFF disables Enable DC Offset Correction (unchecks check box)

1 or ON enables Enable DC Offset Correction (checks check box)

## Returns

<Boolean> 0,1

## Examples

`SENSe:WLAN:AD:ANALYSIS:DC:OFFSET:CORRECTION:STATE 0` will disable the Enable DC Offset Correction check box.

## [SENSe]:WLAN:AD:ANALysis:EQUALizer:CONVergence

Sets or queries the Convergence value (set in the Advanced Params tab of the Settings control panel) for the 802.11ad displays.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

## Conditions

Measurement views: All 802.11ad measurement views.

It is applicable only when ADAPtive is set as the Adaptive Equalizer.

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:CONVergence<value>
```

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:CONVergence?
```

## Arguments

<value>=<NRf>

## Returns

<NRf>

## Examples

`SENSe:WLAN:AD:ANALYSIS:EQUALIZER:CONVERGENCE` might return 2.000E-3, indicating the Convergence value is 2.000E-3.

## **[SENSe]:WLAN:AD:ANALysis:EQUALizer:LENGth**

Sets or queries the adaptive equalizer Length (set in the Advanced Params tab of the Settings control panel) for the 802.11ad displays.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

### **Conditions**

Measurement views: All 802.11ad measurement views

Only applies when ADAPtive is set as the Adaptive Equalizer

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:LENGth <value>
```

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:LENGth?
```

### **Arguments**

<value>=<NR1>

### **Returns**

<NR1>

### **Examples**

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:LENGTH 25 will set the Length to 25.

## **[SENSe]:WLAN:AD:ANALysis:EQUALizer:MODE**

Sets or queries the equalizer mode (set in the Advanced Params tab of the Settings control panel) for the 802.11ad displays.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

### **Conditions**

Measurement views: All 802.11ad measurement views

It is applicable only when ADAPtive is set as the Adaptive Equalizer.

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:MODE {TRA | HOLD}
```

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:MODE?
```

### **Arguments**

TRA will set the mode to Train

HOLD will set the mode to Hold

## Returns

<String> = TRA or HOLD

## Examples

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:MODE TRA sets the equalizer mode to Train.

## [SENSe]:WLAN:AD:ANALYSIS:EQUALizer:RESet (No Query Form)

Resets the equalization value (button in the Advanced Params tab of the Settings control panel) for the 802.11ad displays.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

## Conditions

Measurement views: All 802.11ad measurement views

Only applies when ADAPtive is set as the Adaptive Equalizer

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALYSIS:EQUALizer:RESet
```

## Arguments

None

## Returns

None

## Examples

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:RESET resets the equalization value.

## [SENSe]:WLAN:AD:ANALYSIS:EQUALizer:TAPS

Sets or queries the Taps value in the Advanced Params tab of the Settings control panel for the 802.11ad displays.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

## Conditions

Measurement views: All 802.11ad measurement views

Only applies when ADAPtive is set as the Adaptive Equalizer

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:TAPS <value>  
[SENSe]:WLAN:AD:ANALysis:EQUALizer:TAPS?
```

## Arguments

<value>=<NR1>

## Returns

<NR1>

## Examples

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:TAPS 56 sets the Tap value to 56.

## [SENSe]:WLAN:AD:ANALysis:EQUALizer:TSRatio

Sets or queries the Taps/Symbols value (set in the Advanced Params tab of the Settings control panel) for the 802.11ad displays. (If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

## Conditions

Measurement views: All 802.11ad measurement views

It is applicable only when ADAPtive is set as the Adaptive Equalizer

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:EQUALizer:TSRatio { ONE | TWO | FOUR }  
[SENSe]:WLAN:AD:ANALysis:EQUALizer:TSRatio?
```

## Arguments

ONE , TWO, FOUR

## Returns

<String> = either ONE, TWO, or FOUR

## Examples

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:TSRATIO might return FOUR, indicating the Taps/Symbols value is 4.

## [SENSe]:WLAN:AD:ANALysis:EQUALizer:TYPE

It sets or Queries the Equalizer to OFF, Non Adaptive Equalizer and Adaptive Equalizer in the settings control panel common to all the WLAN 802.11ad displays.

## Conditions

Measurement views: All WLAN 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALYSIS:EQUALIZER:TYPE {NONE | NONA | ADAPtive}
```

```
[SENSe]:WLAN:AD:ANALYSIS:EQUALIZER:TYPE?
```

## Arguments

NONE will set the Equalizer to OFF.

NONA will set the Equalizer to Non Adaptive Equalizer.

ADAPtive will set the Equalizer to Adaptive Equalizer.

## Examples

SENSe:WLAN:AD:ANALYSIS:EQUALIZER:TYPE NONA will set the Equalizer to Non Adaptive Equalizer.

## [SENSe]:WLAN:AD:ANALYSIS:EQUALIZER:TYPE

Sets or queries the Equalizer type in the 802.11ad measurement views.

(If you have SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.)

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALYSIS:EQUALIZER:TYPE { NONE | NONA | ADAPtive }
```

```
[SENSe]:WLAN:AD:ANALYSIS:EQUALIZER:TYPE?
```

## Arguments

NONE will set the Equalizer to OFF

NONA will set the Equalizer to Non Adaptive Equalizer

ADAPtive will set the Equalizer to Adaptive Equalizer

## Returns

<String> NONE or NONA or ADAPtive

## Examples

`SENSe:WLAN:AD:ANALYSIS:EQUALIZER:TYPE ADaptive` will set the equalizer to Adaptive Equalizer.

## **[SENSe]:WLAN:AD:ANALysis:HANDle:MULTi:PATH:CHANnel:STATe**

By enabling this flag, severe multipath can be equalized. By default this flag is set to 0.

### Conditions

Measurement views: All WLAN 802.11ad measurement views

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:ANALysis:HANDle:MULTi:PATH:CHANnel:STATe {0| 1}
```

```
[SENSe]:WLAN:AD:ANALysis:HANDle:MULTi:PATH:CHANnel:STATe ?
```

### Arguments

0 severe multipath cannot be equalized (checkbox is unchecked)

1 severe multipath can be equalized (checkbox is checked)

### Returns

0 or 1, <NRf>

## Examples

`SENSe:WLAN:AD:ANALYSIS:HANDLE:MULTI:PATH:CHANNEL:STATE 1` equalizes the severe multipath.

## **[SENSe]:WLAN:AD:ANALysis:LENGth**

Specifies the length of the analysis period to be used in all the 802.11ad measurement views (set in the Analysis Time tab of the Settings control panel). The analysis length is specified in either symbols or seconds, depending on the Units setting.

### Conditions

Measurement views: All 802.11ad measurement views

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:ANALysis:LENGth <value>
```

```
[SENSe]:WLAN:AD:ANALysis:LENGth?
```

## Arguments

<value>=<NRf>

## Returns

<NRf> = analysis length. The minimum value depends on the modulation type

## Examples

SENSe:WLAN:AD:ANALYSIS:LENGTH 24.7835200000E-6 sets the analysis length to 24.784  $\mu$ s.

## [SENSe]:WLAN:AD:ANALysis:LENGth:ACTual? (Query Only)

Returns the value of the actual analysis length for the 802.11ad measurement views (in the Analysis Time tab of the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

[SENSe]:WLAN:AD:ANALysis:LENGth:ACTual?

## Arguments

None

## Returns

<NRf> = actual analysis length in seconds

## Examples

SENSe:WLAN:AD:ANALYSIS:LENGTH:ACTUAL? might return 2.840E-6, indicating that the actual analysis length is 2.840  $\mu$ s.

## [SENSe]:WLAN:AD:ANALysis:LENGth:AUTO

Enables or disables the Auto checkbox state for analysis length in all the 802.11ad displays (set in the Analysis Time tab of the Settings control panel). This setting determines if the analysis length display is set automatically (enabled) or manually (disabled).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:LENGth:AUTO {0 | OFF | 1 | ON}
```

```
[SENSe]:WLAN:AD:ANALysis:LENGth:AUTO?
```

## Arguments

0 or OFF sets the analysis length to manual (box is unchecked)

1 or ON sets the analysis length to automatic (box is checked)

## Returns

<Boolean> 0,1

## Examples

SENSe:WLAN:AD:ANALYSIS:LENGTH:AUTO ON sets the analysis length value automatically based on the requirements of the selected display.

## [SENSe]:WLAN:AD:ANALysis:OFFSet

Sets or queries the analysis offset value in all the 802.11ad measurement displays (set in the Analysis Time tab in the Settings control panel). The query command returns the result in symbols or seconds, depending on the unit choice selected in the Analysis Time tab of the Settings control panel.

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:ANALysis:OFFSet <value>
```

```
[SENSe]:WLAN:AD:ANALysis:OFFSet?
```

## Arguments

<value>=<NRf>

## Returns

<NRf> = analysis offset

## Examples

SENSe:WLAN:AD:ANALYSIS:OFFSET -1.2E-6 sets the analysis offset to -1.200  $\mu$ s.



## [SENSe]:WLAN:AD:ANALysis:OFFSet:AUTO

Enables or disables the Auto check box for analysis offset in all the 802.11ad measurement views (set in the Analysis Time tab of the Settings control panel). This setting determines if the analysis offset displays is set automatically (enabled) or manually (disabled).

### Conditions

Measurement views: All 802.11ad measurement views

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:ANALysis:OFFSet:AUTO {0 | OFF | 1 | ON}
```

```
[SENSe]:WLAN:AD:ANALysis:OFFSet:AUTO?
```

### Arguments

0 or OFF sets the analysis offset to disabled (box is unchecked)

1 or ON sets the analysis offset to enabled (box is checked)

### Returns

<Boolean> 0,1

### Examples

SENSe:WLAN:AD:ANALYSIS:OFFSET:AUTO 1 sets the analysis offset to automatic.

## [SENSe]:WLAN:AD:ANALysis:SWAP:IQ

Sets or queries whether or not to swap I and Q data.

### Conditions

Measurement views: All WLAN 802.11ad measurement views

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:ANALysis:SWAP:IQ {0 | 1}
```

```
[SENSe]:WLAN:AD:ANALysis:SWAP:IQ ?
```

### Arguments

0 does not swap I and Q data (checkbox is unchecked.)

1 swaps I and Q data (checkbox is checked.)

## Returns

0 or 1, <NR.f>

## Examples

`SENSe:WLAN:AD:ANALYSIS:SWAP:IQ 1` swaps I and Q data.

## [SENSe]:WLAN:AD:ANALysis:TIME:UNITs

Sets or queries the analysis time units for all 802.11ad measurement views (set in the Analysis Time tab of the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AD:ANALysis:TIME:UNITs {SEC, SYMB}`

`[SENSe]:WLAN:AD:ANALysis:TIME:UNITs?`

## Arguments

SEC sets the time units to seconds

SYMB sets the time units to symbols

## Returns

See Arguments

## Examples

`SENSe:WLAN:AD:ANALYSIS:TIME:UNITs SEC` sets the analysis time units to seconds.

## [SENSe]:WLAN:AD(:BANDwidth|:BWIDth):MODE

Sets or queries the measurement BW mode in all the 802.11ad measurement displays (set in the Analysis Params tab in the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AD(:BANDwidth|:BWIDth):MODE { MAN | AUTO | LSP }`

```
[SENSe]:WLAN:AD(:BANDwidth|:BWIDth):MODE?
```

### Arguments

MAN sets the mode to Manual

AUTO sets the mode to Auto

LSP sets the mode to Link to Span

### Returns

See Arguments

### Examples

SENSe:WLAN:AD(:BANDWIDTH|:BWIDTh):MODE LSP will set the Measurement BW mode to Link to Span.

## [SENSe]:WLAN:AD(:BANDwidth|:BWIDth):TINTerval

Sets or queries the Measurement BW in all the 802.11ad measurement displays (set in the Analysis Params tab in the Settings control panel).

### Conditions

Measurement views: All 802.11ad measurement views

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD(:BANDwidth|:BWIDth):TINTerval <value>
```

```
[SENSe]:WLAN:AD(:BANDwidth|:BWIDth):TINTerval?
```

### Arguments

<value>=<NRf>

### Returns

<NRf> = Measurement BW value in Hz

### Examples

SENSe:WLAN:AD(:BANDWIDTH|:BWIDTh):TINTERVAL 5 will set the Measurement BW to 5 Hz.

## [SENSe]:WLAN:AD:EVM:BURSt:COUNT

Sets or queries the total number of bursts to be analyzed (set in the EVM tab of the Settings control panel) for the display.

### Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary view

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:EVM:BURSt:COUNT <value>
```

```
[SENSe]:WLAN:AD:EVM:BURSt:COUNT?
```

## Arguments

<value>=<NR1> Range 1 to 99

## Returns

<NR1> = number of bursts

## Examples

SENSe:WLAN:AD:EVM:BURST:COUNT might return 20, indicating the total number of bursts to be analyzed is 20.

## [SENSe]:WLAN:AD:EVM:BURSt:COUNT:ENABLE

Enables or disables the Max Bursts to Avg check box in the EVM tab of the Settings control panel for the display.

## Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary view

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:EVM:BURSt:COUNT:ENABLE {0 | OFF | 1 | ON }
```

```
[SENSe]:WLAN:AD:EVM:BURSt:COUNT:ENABLE?
```

## Arguments

0 or OFF disables the Max Bursts to Avg (unchecks check box)

1 or ON enables the Max Bursts to Avg (checks check box)

## Returns

<Boolean> 0,1

## Examples

SENSe:WLAN:AD:EVM:BURST:COUNT:ENABLE might return 1, indicating the Max Bursts to Avg option is enabled.

## [SENSe]:WLAN:AD:EVM:DATA:ANALyze:ALL:SYMBOLs

Enables or disables the Analyze All check box in the EVM tab of the Settings control panel for the display. When the Analyze All check box is enabled, all the data symbols are analyzed and the data start value is always 0.

### Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary view

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:EVM:DATA:ANALyze:ALL:SYMBOLs { 0 | OFF | 1 | ON }
```

```
[SENSe]:WLAN:AD:EVM:DATA:ANALyze:ALL:SYMBOLs?
```

### Arguments

0 or OFF disables Analyze All (unchecks check box)

1 or ON enables Analyze All (checks check box)

### Returns

<Boolean> 0,1

### Examples

SENSe:WLAN:AD:EVM:DATA:ANALYZE:ALL:SYMBOLS might return 0, indicating that the Data start and Data length values are considered for current analysis.

## [SENSe]:WLAN:AD:EVM:DATA:LENGth

Sets or queries the length value (set in the EVM tab of the Settings control panel) in the display.

### Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary view

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AD:EVM:DATA:LENGth <value>
```

```
[SENSe]:WLAN:AD:EVM:DATA:LENGth?
```

### Arguments

<value>=<NR1>

## Returns

<NR1> = length value

## Examples

`SENSe:WLAN:AD:EVM:DATA:LENGTH 1000`, indicating that analysis should happen for 1000 symbols in the acquired data region.

## [SENSe]:WLAN:AD:EVM:DATA:START

Sets or queries the start value (set in the EVM tab of the Settings control panel) for the display.

## Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary view

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AD:EVM:DATA:START <value>`

`[SENSe]:WLAN:AD:EVM:DATA:START?`

## Arguments

<value>=<NR1>

## Returns

<NR1> = start value

## Examples

`SENSe:WLAN:AD:EVM:DATA:START 5` indicates that analysis should begin at the 5th symbol.

## [SENSe]:WLAN:AD:EVM:UNITs

Sets or queries the EVM units in the EVM tab of the Settings control panel.

## Conditions

Measurement views: 802.11ad Constellation, EVM vs Time, and Summary views

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AD:EVM:UNITs { DB | PERC }`

`[SENSe]:WLAN:AD:EVM:UNITs?`

## Arguments

DB will set the EVM units to decibels PERC will set the EVM units to percent

## Returns

See Arguments

## Examples

`SENSe:WLAN:AD:EVM:UNITS PERC` sets the EVM units to percent.

## [SENSe]:WLAN:AD:FILTer:ALPHa

Sets or queries the Filter Parameter for all 802.11ad measurement views, which is applicable when RRC is set as the measurement filter type (in the Modulation Params tab of the Settings control panel).

## Conditions

Measurement views: All WLAN 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:FILTer:ALPHa <value>
```

```
[SENSe]:WLAN:AD:FILTer:ALPHa?
```

## Arguments

<value>=<NRf>

## Returns

<NRf> = filter parameter

## Examples

`SENSe:WLAN:AD:FILTer:ALPHa 254.0000000000E-3` will set the filter parameter to 0.254.

## [SENSe]:WLAN:AD:FILTer:MEASurement

Sets or queries the measurement filter in all the 802.11ad measurement views (set in the Modulation Params tab of the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:FILTer:MEASurement { OFF | RRC }  
[SENSe]:WLAN:AD:FILTer:MEASurement?
```

## Arguments

OFF sets the filter to None

RRC sets the filter to Root raised cosine

## Returns

See Arguments

## Examples

SENSe:WLAN:AD:FILTER:MEASUREMENT might return RRC, indicating that the measurement filter is set to Root Raised Cosine.

## [SENSe]:WLAN:AD:FREQuency:ERRor

Sets or queries the frequency error value in all the 802.11ad measurement views (set in the Analysis Params tab of the Settings control panel).

## Conditions

Measurement views: All WLAN 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AD:FREQuency:ERRor <value>  
[SENSe]:WLAN:AD:FREQuency:ERRor?
```

## Arguments

<value>=<NRf>

## Returns

<NRf> = frequency error in Hz

## Examples

SENSe:WLAN:AD:FREQUENCY:ERROR 6 will set the frequency error to 6.000 Hz.

## [SENSe]:WLAN:AD:FREQuency:ERRor:AUTO

Enables or disables the Auto check box state for frequency error in all the 802.11ad displays in the Analysis Params tab of the Settings control panel. This setting determines whether the frequency error is set automatically (enabled) or manually (disabled).



## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSE]:WLAN:AD:FREQUENCY:ERROR:AUTO { 0 | OFF | 1 | ON }
```

```
[SENSE]:WLAN:AD:FREQUENCY:ERROR:AUTO?
```

## Arguments

OFF or 0 disables Frequency Error Auto (Manual mode) ON or 1 enables Frequency Error Auto (Auto mode)

## Returns

<Boolean> 0, 1

## Examples

SENSE:WLAN:AD:FREQUENCY:ERROR:AUTO OFF disables the automatic mode for frequency error.

## [SENSE]:WLAN:AD:MODULATION

Sets or queries the standard modulation type in all the 802.11ad measurement views (set in the Modulation Params tab of the Settings control panel).

## Conditions

Measurement views: All 802.11ad measurement views

## Group

Sense commands

## Syntax

```
[SENSE]:WLAN:AD:MODULATION {SC,CPHY}
```

```
[SENSE]:WLAN:AD:MODULATION?
```

## Arguments

SC will set the standard to Single Carrier CPHY will set the standard to control Phy

## Returns

See Arguments

## Examples

SENSE:WLAN:AD:MODULATION SC will set the standard modulation type to Single Carrier.

## **[SENSe]:WLAN:AD:SUMMARY:CLEar:RESults (No Query Form)**

Clears all scalar results in the display.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AD:SUMMARY:CLEar:RESults
```

### **Arguments**

None

### **Returns**

None

### **Examples**

`SENSe:WLAN:AD:SUMMARY:CLEAR:RESULTS` clears the result in the 802.11ad Summary display.

## **[SENSe]:WLAN:AD:SUMMARY:LIMIT:EVM**

Sets or queries EVM limit value (set in the Limit tab in the Settings control panel) for the display. This value is the threshold for pass or fail.

### **Conditions**

Measurement view: 802.11ad Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AD:SUMMARY:LIMIT:EVM <Arg1,Arg2>
```

```
[SENSe]:WLAN:AD:SUMMARY:LIMIT:EVM?
```

### **Arguments**

<Arg1> = OFF or 0 (disables EVM limit) or ON or 1 (enables EVM limit)

<Arg2> = limit value (Range: 0 to -160)

### **Returns**

<Boolean> 0,1 and <NRf> (0 to -160)

## Examples

`SENSe:WLAN:AD:SUMMARY:LIMIT:EVM 1,-30` enables the EVM limit and sets the limit to -30.

## [SENSe]:WLAN:AD:SUMMARY:LIMIT:FREQUENCY:ERROR

Sets or queries the frequency error limit (set in the Limit tab of the Settings control panel) for the display. This value is the threshold for pass or fail.

### Conditions

Measurement view: 802.11ad Summary

### Group

Sense commands

### Syntax

`[SENSe]:WLAN:AD:SUMMARY:LIMIT:FREQUENCY:ERROR <Arg1,Arg2>`

`[SENSe]:WLAN:AD:SUMMARY:LIMIT:FREQUENCY:ERROR?`

### Arguments

`<Arg1>` = OFF or 0 (disables Freq Error Limit) or ON or 1 (enables Freq Error Limit)

`<Arg2>` = limit value

### Returns

`<Boolean>` 0,1

## Examples

`SENSe:WLAN:AD:SUMMARY:LIMIT:FREQUENCY:ERROR 0,60` disables the Freq Error Limit and sets the limit to 60.

## [SENSe]:WLAN:ANALYSIS:LENGTH

Specifies the length of the analysis period to use in the WLAN analysis time settings. Analysis length is specified in either symbols or seconds, depending on the Units setting. Use `[SENSe]:WLAN:UNIT:TIME` to set the units.

Programming a specified length sets `[SENSe]:WLAN:ANALYSIS:AUTO` to OFF.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

`[SENSe]:WLAN:ANALYSIS:LENGTH <Nrf>`

[SENSe]:WLAN:ANALYsis:LENGth?

### Arguments

Floating point number that specifies the analysis length.

Range: the minimum value depends on the modulation type.

### Examples

SENSe:WLAN:ANALYSIS:LENGTH 25.625e-6 sets the analysis length to 25.625  $\mu$ s.

## [SENSe]:WLAN:ANALYsis:LENGth:ACTual? (Query Only)

Returns the value of the actual analysis length for the WLAN analysis time settings.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

[SENSe]:WLAN:ANALYsis:LENGth:ACTual?

### Returns

Floating point number that represents the actual analysis length in seconds.

### Examples

SENSe:WLAN:ANALYSIS:LENGTH:ACTUAL? might return 25.625E-6, indicating that the actual analysis length is 25.625  $\mu$ s.

## [SENSe]:WLAN:ANALYsis:LENGth:AUTO

Sets the analysis length value based on the requirements of the selected display. To set the analysis length manually, use the command.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

[SENSe]:WLAN:ANALYsis:LENGth:AUTO {0|1|OFF|ON}

[SENSe]:WLAN:ANALYsis:LENGth:AUTO?

## Arguments

ON or 1 sets the analysis length value based on the requirements of the selected display.

OFF or 0 turns this function off.

## Examples

`SENSe:WLAN:ANALYSIS:LENGTH:AUTO ON` sets the analysis length value automatically based on the requirements of the selected display.

## [SENSe]:WLAN:ANALysis:OFFSet

Specifies or queries the analysis offset in the symbol interval in WLAN analysis parameters. This value is a percentage of the Guard Interval. The allowable range is -100% to 0%. -100% positions the FFT to start at the beginning of the Guard Interval, 0% positions it to start at the end of the Guard Interval. The default value is -50% which usually gives the best measurement results.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:ANALysis:OFFSet <NR1>
```

```
[SENSe]:WLAN:ANALysis:OFFSet?
```

## Arguments

Integer that represents the analysis offset in the symbol interval, in percent terms.

## Examples

`SENSe:WLAN:ANALYSIS:OFFSET -50` sets the analysis offset in the symbol interval to -50%.

## [SENSe]:WLAN:AY:ANALysis:BLOCKwise:CORRection:STATe

Set or query the state of the block wise data analysis (disabled or enabled).

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:BLOCKwise:CORRection:STATe { OFF | ON | 0 | 1 }
```

[SENSe]:WLAN:AY:ANALysis:BLOCKwise:CORRection:STATe?

### Arguments

0 or OFF disables block wise correction.

1 or ON enables block wise correction.

### Returns

0 or OFF means block wise correction is turned off.

1 or ON means block wise correction is turned on.

### Examples

SENSe:WLAN:AY:ANAL:BLOC:CORR:STAT 1 turns on block wise correction.

SENSe:WLAN:AY:ANAL:BLOC:CORR:STAT? may return 0, which means block wise correction is turned off.

## [SENSe]:WLAN:AY:ANALysis:EQUALizer:CONVergence

Set or query the Convergence value (set in the Advanced Parameters tab of the Settings control panel) for the 802.11ay displays.

### Conditions

Measurement views: All 802.11ay

It is applicable only when Adaptive is set as the Adaptive Equalizer.

### Group

Sense commands

### Syntax

[SENSe]:WLAN:AY:ANALysis:EQUALizer:CONVergence <value>

[SENSe]:WLAN:AY:ANALysis:EQUALizer:CONVergence?

### Arguments

<value>::=<NRf> is the Convergence value.

### Returns

<NRf>

### Examples

SENSe:WLAN:AY:ANAL:EQUA:CONV 2.000e-3 sets the Convergence value to 2.000e-3.

SENSe:WLAN:AY:ANAL:EQUA:CONV? may return 2.000e-3, indicating the Convergence value is 2.000e-3.

## [SENSe]:WLAN:AY:ANALysis:DC:OFFSet:CORRection:STATe

Set or query the state of the Enable DC Offset Correction checkbox (checked or unchecked) in the Settings control panel common to all the WLAN 802.11ay displays.

### Conditions

Measurement views: All WLAN 802.11ay

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:ANALysis:DC:OFFSet:CORRection:STATe { OFF | ON | 0 | 1 }
```

```
[SENSe]:WLAN:AY:ANALysis:DC:OFFSet:CORRection:STATe?
```

### Arguments

0 or OFF turns off Enable DC Offset Correction.

1 or ON turns on Enable DC Offset Correction.

### Returns

0 or OFF means that Enable DC Offset Correction is disabled.

1 or ON means that Enable DC Offset Correction is enabled.

### Examples

`SENSe:WLAN:AY:ANAL:DC:OFFS:CORR:STAT 0` unchecks the Enable DC Offset Correction checkbox.

`SENSe:WLAN:AY:ANAL:DC:OFFS:CORR:STAT?` may return 1, which means that the Enable DC Offset Correction checkbox is checked

## [SENSe]:WLAN:AY:ANALysis:EQUALizer:LENGth

Set or query the Adaptive Equalizer Length (set in the Advanced Parameters tab of the Settings control panel) for the 802.11ay displays.

If you use SignalVu for scopes, this command is only available for the SignalVu version 3.10.0030 and above.)

### Conditions

Measurement views: All 802.11ay

Only applies when adaptive is set as the adaptive equalizer.

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:LENGth <value>
```

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:LENGth?
```

## Arguments

<value> ::= <NR1> is the Adaptive Equalizer Length value.

## Returns

<NR1>

## Examples

SENSe:WLAN:AY:ANAL:EQUA:LENG 25 sets Adaptive Equalizer Length to 25.

SENSe:WLAN:AY:ANAL:EQUA:LENG? may return 25, which indicates that Adaptive Equalizer Length is 25.

## [SENSe]:WLAN:AY:ANALysis:EQUALizer:MODE

Set or query the Equalizer Mode (set in the Advanced Parameters tab of the Settings control panel) for 802.11ay displays.

If you use SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.

## Conditions

Measurement views: All 802.11ay

It is applicable only when Adaptive is set as the Adaptive Equalizer.

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:MODE { TRA | HOLD }
```

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:MODE?
```

## Arguments

TRA sets mode to Train.

HOLD sets mode to Hold.

## Returns

<string> = TRA or HOLD

## Examples

SENSe:WLAN:AY:ANAL:EQUA:MODE TRA sets the Equalizer Mode to Train.

SENSe:WLAN:AY:ANAL:EQUA:MODE? may return HOLD, which indicates that the Equalizer Mode is Hold.

## [SENSe]:WLAN:AY:ANALysis:EQUALizer:RESet (No Query Form)

Reset the Equalization value (button in the Advanced Parameters tab of the Settings control panel) for the 802.11ay displays.

If you use SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.



## Conditions

Measurement views: All 802.11ay

It is applicable only when Adaptive is set as the Adaptive Equalizer.

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:EQUAlizer:RESet
```

## Arguments

None

## Examples

SENSe:WLAN:AY:ANAL:EQUA:RES resets the Equalization value.

## [SENSe]:WLAN:AY:ANALysis:EQUAlizer:TAPS

Set or query the Taps value in the Advanced Parameters tab of the Settings control panel for the 802.11ay displays.

If you use SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.

## Conditions

Measurement views: All 802.11ay

It is applicable only when Adaptive is set as the Adaptive Equalizer.

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:EQUAlizer:TAPS <value>
```

```
[SENSe]:WLAN:AY:ANALysis:EQUAlizer:TAPS?
```

## Arguments

<value>: :=<NR1> is the Taps value.

## Returns

<NR1>

## Examples

SENSe:WLAN:AY:ANAL:EQUA:TAPS 56 sets the Taps value to 56.

SENSe:WLAN:AY:ANAL:EQUA:TAPS? may return 56, which means that the Taps value is 56.

## **[SENSe]:WLAN:AY:ANALysis:EQUALizer:TSRatio**

Set or query the Taps/Symbols value (set in the Advanced Parameters tab of the Settings control panel) for the 802.11ay displays.

If you use SignalVu for scopes, this command is only available for SignalVu version 3.10.0030 and above.

### **Conditions**

Measurement views: All 802.11ay

It is applicable only when Adaptive is set as the Adaptive Equalizer.

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:TSRatio { ONE | TWO | FOUR }
```

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:TSRatio?
```

### **Arguments**

ONE, TWO, FOUR

### **Returns**

<string> = ONE, TWO, or FOUR

### **Examples**

`SENSe:WLAN:AY:ANAL:EQUA:TSR ONE` sets the Taps/Symbols value to 1.

`SENSe:WLAN:AY:ANAL:EQUA:TSR?` may return FOUR, indicating the value of Taps/Symbols is 4.

## **[SENSe]:WLAN:AY:ANALysis:EQUALizer:TYPE**

Set or query whether the Equalizer is set to OFF, Nonadaptive (NONA), or Adaptive (ADAPTive) in the settings control panel common to all the WLAN 802.11ay displays.

### **Conditions**

Measurement views: All WLAN 802.11ay

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:TYPE { NONE | NONA | ADAPTive }
```

```
[SENSe]:WLAN:AY:ANALysis:EQUALizer:TYPE?
```

### **Arguments**

NONE sets the equalizer to OFF.

`NONA` sets the equalizer to nonadaptive.

`ADAPtive` sets the equalizer to adaptive.

## Returns

`NONE` means that the equalizer is set to OFF.

`NONA` means that the equalizer is set to Nonadaptive.

`ADAPtive` means that the equalizer is set to Adaptive.

## Examples

`SENSe:WLAN:AY:ANAL:EQUAL:TYPE ADAPtive` sets the equalizer to Adaptive

`SENSe:WLAN:AY:ANAL:EQUAL:TYPE?` may return `NONE`, which means that the equalizer is set to OFF.

## [SENSe]:WLAN:AY:ANALysis:LENGth

Set or query the length value of the analysis period to be used in the WLAN 802.11ay analysis time settings.

The analysis length is specified in symbols or seconds, depending on the units setting.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:ANALysis:LENGth <NRf>`

`[SENSe]:WLAN:AY:ANALysis:LENGth?`

## Arguments

`<NRf>` is the floating point number that specifies the analysis length.

Range: the minimum value depends on the modulation type. Units:  $\mu$ s or s (seconds)

## Returns

`<NRf>`

## Examples

`SENSe:WLAN:AY:ANAL:LENG 24.7835200000e-6` sets the analysis length 24.784  $\mu$ s.

`SENSe:WLAN:AY:ANAL:LENG?` may return 24.7835200000e-6, which means that the analysis length is 24.784  $\mu$ s.

## [SENSe]:WLAN:AY:ANALysis:LENGth:ACTual? (Query Only)

Query the value of the actual analysis length for the WLAN 802.11ay analysis time settings.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:LENGth:ACTual?
```

## Returns

<NRf>

Units:  $\mu$ s or s (seconds)

## Examples

SENSe:WLAN:AY:ANAL:LENG:ACT? may return 2.840e-6, indicating that the actual analysis length is 2.840  $\mu$ s.

## [SENSe]:WLAN:AY:ANALysis:LENGth:AUTO

Set or query the state of the analysis length (enabled or disabled) based on the requirements of the selected display.

To set the analysis length value manually, use this command.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:ANALysis:LENGth:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:WLAN:AY:ANALysis:LENGth:AUTO?
```

## Arguments

ON or 1 enables analysis length.

OFF or 0 disables analysis length.

## Returns

ON or 1 sets analysis length to automatic (box is checked).

OFF or 0 sets analysis length to manual (box is unchecked).

## Examples

SENSe:WLAN:AY:ANAL:LENG:AUTO ON sets analysis length automatically.

`SENSe:WLAN:AY:ANAL:LENG:AUTO?` may return OFF, which means analysis length is turned off.

## **[SENSe]:WLAN:AY:ANALysis:OFFSet**

Set or query the value of the Analysis Offset for all the WLAN 802.11ay displays. The value is located under the Settings control panel in all WLAN 802.11ay displays.

The query returns the result in symbols or seconds, based on the unit choice selected in the analysis time tab in the WLAN 802.11ay settings control panel.

### **Conditions**

Measurement views: All WLAN 802.11ay

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:ANALysis:OFFSet <value>
```

```
[SENSe]:WLAN:AY:ANALysis:OFFSet?
```

### **Arguments**

`<value>::=<Nrf>` sets the analysis offset in the Analysis Time tab of the WLAN 802.11ay Settings control panel.

Units:  $\mu$ s or s (seconds)

### **Returns**

`<Nrf>`

### **Examples**

`SENSe:WLAN:AY:ANAL:OFFS -1.2e-6` sets the Analysis Offset to  $-1.200 \mu$ s.

`SENSe:WLAN:AY:ANAL:OFFS?` may return  $-1.2e-6$ , which means that the analysis offset is  $-1.200 \mu$ s.

## **[SENSe]:WLAN:AY:ANALysis:OFFSet:AUTO**

Sets or query the state of the Auto checkbox (enabled or disabled) for the analysis offset.

This setting determines if the analysis offset displays is set automatically or manually.

### **Conditions**

Measurement views: All WLAN 802.11ay

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:ANALysis:OFFSet:AUTO { OFF | ON | 0 | 1 }
```

```
[SENSe]:WLAN:AY:ANALysis:OFFSet:AUTO?
```

### Arguments

0 or OFF sets the analysis offset to manual (box is unchecked).

1 or ON sets the analysis offset to automatic (box is checked).

### Returns

0 or OFF means that the analysis offset is manual.

1 or ON means that the analysis offset is automatic.

### Examples

SENSe:WLAN:AY:ANAL:OFFS:AUTO 1 sets the analysis offset to automatic.

SENSe:WLAN:AY:ANAL:OFFS:AUTO? may return 0, which means that the analysis offset to manual.

## [SENSe]:WLAN:AY:ANALysis:SFO:TRACk:STATe

Set or query the state of the SFO Tracker. When enabled, the SFO Tracker estimates and corrects Symbol Timing/Frequency errors for each Data/GI block. When disabled, it performs a single shot correction for the entire packet, which is based on the value estimated from the STF Region.

### Conditions

Measurement views: All WLAN 802.11ay

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:ANALysis:SFO:TRACk:STATe { OFF | ON | 0 | 1 }
```

### Arguments

0 or OFF disables the SFO Tracker.

1 or ON enables the SFO Tracker.

### Returns

0 or OFF means the SFO Tracker is turned off.

1 or ON means the SFO Tracker is turned on.

### Examples

SENSe:WLAN:AY:ANAL:SFO:TRAC:STAT 0 turns off the SFO Tracker, performing a single shot correction for the entire data packet.

SENSe:WLAN:AY:ANAL:SFO:TRAC:STAT? may return 1, which indicates that the SFO Tracker is turned on, correcting Symbol Timing and Frequency errors for each Data/GI block in the packet.

## [SENSe]:WLAN:AY:ANALysis:SWAP:IQ

Set or query the state of the Maximum Bursts to Average option (enabled or disabled) in the WLAN 802.11ay Summary setting control panel.

### Conditions

Measurement views: WLAN 802.11ay Summary

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:ANALysis:SWAP:IQ { OFF | ON | 0 | 1 }
```

```
[SENSe]:WLAN:AY:ANALysis:SWAP:IQ?
```

### Arguments

OFF or 0 disables Maximum Bursts to Average.

1 or ON enables Maximum Bursts to Average.

### Returns

0 or OFF means Maximum Bursts to Average is turned off.

1 or ON means Maximum Bursts to Average is turned on.

### Examples

`SENSe:WLAN:AY:ANAL:SWAP:IQ 0` disables the IQ Swap for the WLAN 802.11ay Summary display.

`SENSe:WLAN:AY:ANAL:SWAP:IQ?` may return 1, which indicates that the IQ Swap is enabled for the WLAN 802.11ay Summary display.

## [SENSe]:WLAN:AY:ANALysis:TIME:UNITs

Set or query the type of time units in the WLAN 802.11ay settings.

### Conditions

Measurement views: All WLAN 802.11ay

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:ANALysis:TIME:UNITs { SEC | SYMB }
```

```
[SENSe]:WLAN:AY:ANALysis:TIME:UNITs?
```

## Arguments

SEC sets time units to seconds.

SYMB sets time units to symbols.

## Returns

SEC means that time units are set to seconds.

SYMB means that time units are set to symbols.

## Examples

SENSE:WLAN:AY:ANAL:TIM:UNIT SEC sets time units to seconds.

SENSE:WLAN:AY:ANAL:TIM:UNIT? may return SYMB, which means that time units are set to symbols.

## [SENSE]:WLAN:AY:ANALysis:TRACk:EQ:STATE

Set or query the state of the LMS Tracking Equalizer. When enabled, it performs LMS Tracking Equalization for each Data/GI block. When disabled, it provides a single shot tracking and equalization for the entire packet.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSE]:WLAN:AY:ANALysis:TRACk:EQ:STATE { OFF | ON | 0 | 1 }
```

```
[SENSE]:WLAN:AY:ANALysis:TRACk:EQ:STATE?
```

## Arguments

0 or OFF disables the LMS Tracking Equalizer.

1 or ON enables the LMS Tracking Equalizer.

## Returns

0 or OFF means the LMS Tracking Equalizer is turned off.

1 or ON means the LMS Tracking Equalizer is turned on.

## Examples

SENSE:WLAN:AY:ANAL:TRAC:EQ:STAT 1 turns on the LMS Tracking Equalizer for each Data/GI block.

SENSE:WLAN:AY:ANAL:TRAC:EQ:STAT? may return 0, which indicates that the LMS Tracking Equalizer is turned off.

## [SENSE]:WLAN:AY(:BANDwidth|:BWIDth):MODE

Set or query the mode of the Measurement BW in the Settings control panel common to all WLAN 802.11ay displays.



## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY(:BANDwidth|:BWIDth):MODE { MAN | AUTO | LSP}
```

```
[SENSe]:WLAN:AY(:BANDwidth|:BWIDth):MODE?
```

## Arguments

Select one of the below arguments:

Argument	Description
MANual	Allows manual selection of the measurement bandwidth. A typical use is setting a narrow measurement bandwidth on a small range of frequencies to examine one signal, while rejecting the others that are present in a spectrum graph.
AUTO	Automatically selects a measurement bandwidth. The measurement chooses a bandwidth, based on other parameter settings, such as symbol rate and modulation type.
LSP	Sets the Measurement BW to Link to Span. This allows you to use the spectrum display to tune the frequency, view the signal, and then use span control to set the measurement bandwidth. This mode emulates legacy instruments.

## Returns

See Arguments.

## Examples

`SENSe:WLAN:AY:BAND:MODE LSP` sets the Measurement BW mode to Link to Span.

`SENSe:WLAN:AY:BAND:MODE?` may return AUTO, which means that the Measurement BW Mode automatically selects a measurement bandwidth.

## [SENSe]:WLAN:AY(:BANDwidth|:BWIDth):TINTerval

Set or query the value of the Measurement Bandwidth (BW) present in the settings control panel common to all the WLAN 802.11ay displays.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY(:BANDwidth|:BWIDth):TINTerval <value>  
[SENSe]:WLAN:AY(:BANDwidth|:BWIDth):TINTerval?
```

## Arguments

<value>::=<NRf> specifies the measurement bandwidth.

Range: 1 to 100 Hz

## Returns

<NR3>

## Examples

SENSe:WLAN:AY:BAND:BWID:TINT 5 sets the Measurement Bandwidth value to 5 Hz.

SENSe:WLAN:AY:BAND:BWID:TINT? may return 22, which indicates that the Measurement Bandwidth value is 22 Hz.

## [SENSe]:WLAN:AY:CHbw

Set or query the type of channel bandwidth present in the settings control panel common to all WLAN 802.11ay displays.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:CHbw { BW1X | BW2X }  
[SENSe]:WLAN:AY:CHbw?
```

## Arguments

BW1X sets the channel bandwidth to 2.16 GHz.

BW2X sets the channel bandwidth to 4.32 GHz.

Units: GHz

## Returns

BW1X means that the channel bandwidth is 2.16 GHz.

BW2X means that the channel bandwidth is 4.32 GHz.

## Examples

SENSe:WLAN:AY:CH BW1X sets the channel bandwidth to 2.16 GHz.

SENSe:WLAN:AY:CH? may return BW2X, which indicates that the channel bandwidth is 4.32 GHz.

## [SENSe]:WLAN:AY:DUPBand

Set or query the band (Upper or Lower) that is showing the results when a nonEDMG Duplicate mode is detected with 11ay SC 4.32 GHz BW settings.

### Conditions

Measurement views: All WLAN 802.11ay

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:DUPBand { LOWer | UPPer }
[SENSe]:WLAN:AY:DUPBand?
```

### Arguments

LOWer sets the band to 2G-L for all open displays and updates the results.

UPPer sets the band to 2G-U for all open displays and updates the results.

### Returns

LOWer means that the band is set to 2G-L for all open displays

UPPer sets the band to 2G-U for all open displays.

### Examples

SENSe:WLAN:AY:DUPB UPPer sets the band to 2G-U for all open displays.

SENSe:WLAN:AY:DUPB? may return LOWer, which indicates that 2G-L band is set for all open displays.

## [SENSe]:WLAN:AY:EVM:BURSt:COUNT

Set or query the total burst count to be analyzed.

### Conditions

Measurement views: WLAN 802.11ay Summary

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:AY:EVM:BURSt:COUNT <value>
[SENSe]:WLAN:AY:EVM:BURSt:COUNT?
```

### Arguments

<value>::=<NR1> is the total burst count.

## Returns

<NR1>

## Examples

`SENSE:WLAN:AY:EVM:BURSt:COUN 20` sets the total burst count to be analyzed to 20.

`SENSE:WLAN:AY:EVM:BURSt:COUN?` may return 20, which indicates that the total burst count to be analyzed is 20.

## [SENSe]:WLAN:AY:EVM:BURSt:COUNT:ENABLE

Set or query the state of the Maximum Bursts to Average option (enabled or disabled) in the WLAN 802.11ay Summary setting control panel.

## Conditions

Measurement views: WLAN 802.11ay Summary

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:EVM:BURSt:COUNT:ENABLE { OFF | ON | 0 | 1 }`

`[SENSe]:WLAN:AY:EVM:BURSt:COUNT:ENABLE?`

## Arguments

0 or OFF disables Maximum Bursts to Average.

1 or ON enables Maximum Bursts to Average .

## Returns

0 or OFF means that Maximum Bursts to Average is turned off.

1 or ON means that Maximum Bursts to Average is turned on.

## Examples

`SENSE:WLAN:AY:EVM:BURS:COUN:ENAB 0` turns off Maximum Bursts to Average option.

`SENSE:WLAN:AY:EVM:BURS:COUN:ENAB?` may return 1, which indicates that Maximum Bursts to Average option is turned on.

## [SENSe]:WLAN:AY:EVM:DATA:ANALyze:ALL:SYMBOLs

Set or query whether the Analyze All option is enabled or disabled. When enabled, it allows all available data symbols to be analyzed.

Data Start and Data Length values are considered for current analysis. When using standard preset, it defaults to unchecked mode.

## Conditions

Measurement views: WLAN 802.11ay Summary

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:EVM:DATA:ANALyze:ALL:SYMBols { OFF | ON | 0 | 1 }
```

```
[SENSe]:WLAN:AY:EVM:DATA:ANALyze:ALL:SYMBols?
```

## Arguments

0 or OFF disables Analyze All.

1 or ON enables Analyze All.

## Returns

0 or OFF means that Analyze All is turned off.

1 or ON means that Analyze All is turned on.

## Examples

`SENSe:WLAN:AY:EVM:DATA:ANAL:ALL:SYMB 1` enables data start and data length values to be considered for current analysis.

`SENSe:WLAN:AY:EVM:DATA:ANAL:ALL:SYMB?` may return 0, indicating that data start and data length values are considered for current analysis.

## [SENSe]:WLAN:AY:EVM:DATA:LENGth

Set or query whether the Analyze All checkbox is checked or unchecked. When enabled, all data symbols are analyzed and the data start value is always 0.

## Conditions

Measurement views: WLAN 802.11ay Constellation, WLAN 802.11ay EVM versus Time and WLAN 802.11ay Summary displays

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:EVM:DATA:LENGth <value>
```

```
[SENSe]:WLAN:AY:EVM:DATA:LENGth?
```

## Arguments

`<value>::=<NR1>` is the number of symbols to be analyzed.

## Returns

`<NR1>`

## Examples

`SENSe:WLAN:AY:EVM:DATA:LENG 1000` sets the analysis to occur for 1000 symbols.

`SENSe:WLAN:AY:EVM:DATA:LENG?` may return 1000, which means that the analysis occurs for 1000 symbols.

## [SENSe]:WLAN:AY:EVM:DATA:START

Set or query whether the Analyze All checkbox is checked or unchecked. When enabled, all data symbols are analyzed and the data start value is always 0.

## Conditions

Measurement views: WLAN 802.11ay Constellation, WLAN 802.11ay EVM versus Time and WLAN 802.11ay Summary displays

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:EVM:DATA:START <value>`

`[SENSe]:WLAN:AY:EVM:DATA:START?`

## Arguments

`<value>::=<NRf>` is the symbol number.

## Returns

`<NRf>`

## Examples

`SENSe:WLAN:AY:EVM:DATA:STAR 5` sets data analysis to begin at the 5th symbol.

`SENSe:WLAN:AY:EVM:DATA:STAR?` may return 5, which indicates that data analysis begins at the 5th symbol.

## [SENSe]:WLAN:AY:EVM:UNITs

Set or query the type of EVM units (dB or percent).

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:EVM:UNITs { DB | PERC }`

`[SENSe]:WLAN:AY:EVM:UNITs?`

## Arguments

DB sets EVM units to dB.

PERC sets EVM units to %.

## Returns

DB means that EVM units are in dB.

PERC means that EVM units are in %.

## Examples

`SENSe:WLAN:AY:EVM:UNIT PERC` sets EVM units to percent (%).

`SENSe:WLAN:AY:EVM:UNIT?` may return DB, which indicates that EVM unit are in dB.

## [SENSe]:WLAN:AY:FILTer:ALPHa

Set or query the value of the filter parameter, which is applicable when RRC is set as the measurement filter type.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:FILTer:ALPHa <value>`

`[SENSe]:WLAN:AY:FILTer:ALPHa?`

## Arguments

`<value>::=<NRf>` is the filter parameter value.

## Returns

`<NRf>`

## Examples

`SENSe:WLAN:AY:FILT:ALPH 254.000000000e-3` sets the filter parameter value to 0.254.

`SENSe:WLAN:AY:FILT:ALPH?` may return `254.000000000e-3`, which means that the filter parameter value is `254.000000000e-3`.

## [SENSe]:WLAN:AY:FILTer:MEASurement

Set or query type of the WLAN 802.11ay measurement filter setting.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:FILTer:MEASurement { OFF | RRC }
```

```
[SENSe]:WLAN:AY:FILTer:MEASurement?
```

## Arguments

OFF sets the measurement filter settings to None.

RRC sets the measurement filter settings to Root Raised Cosine.

## Returns

OFF means that the measurement filter settings is set to None.

RRC means that the measurement filter settings is set to Root Raised Cosine.

## Examples

`SENSe:WLAN:AY:FILT:MEAS OFF` sets the measurement filter settings to None.

`SENSe:WLAN:AY:FILT:MEAS?` may return RRC, which means that the measurement filter settings is Root Raised Cosine.

## [SENSe]:WLAN:AY:FREQuency:ERRor

Set or query the value of the frequency error present in the Settings control panel common to all WLAN 802.11ay displays.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:AY:FREQuency:ERRor <value>
```

```
[SENSe]:WLAN:AY:FREQuency:ERRor?
```

## Arguments

`<value>::=<NRf>` is the frequency error value.

Range: -2.5 to 2.5 Hz

## Returns

`<NRf>`



## Examples

`SENSe:WLAN:AY:FREQ:ERR 6` sets the frequency error value to 6.000 Hz.

`SENSe:WLAN:AY:FREQ:ERR?` may return 5, which means that the frequency error value is 5.000 Hz.

## [SENSe]:WLAN:AY:FREQuency:ERRor:AUTO

Set or query the state of the Auto checkbox (enabled or disabled) for frequency errors in the settings control panel common to all WLAN 802.11ay displays.

This setting determines if the frequency error is set automatically or manually.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:FREQuency:ERRor:AUTO { OFF | ON | 0 | 1 }`

`[SENSe]:WLAN:AY:FREQuency:ERRor:AUTO?`

## Arguments

OFF or 0 disables the automatic frequency error mode (and sets the manual mode).

ON or 1 enables the automatic frequency error (and sets the auto mode).

## Returns

OFF or 0 means that the automatic frequency error mode is disabled.

ON or 1 means that the automatic frequency error is enabled.

## Examples

`SENSe:WLAN:AY:FREQ:ERR:AUTO OFF` disables the frequency error mode and sets the manual mode.

`SENSe:WLAN:AY:FREQ:ERR:AUTO?` may return ON, which means that the frequency error mode is automatic.

## [SENSe]:WLAN:AY:MODUlation

Set or query the type of modulation present in the settings control panel that is common to all WLAN 802.11ay displays.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

[SENSe]:WLAN:AY:MODulation AYSC

[SENSe]:WLAN:AY:MODulation?

## Arguments

AYSC sets the standard to 802.11ay Single Carrier.

## Returns

AYSC

## Examples

SENSe:WLAN:AY:MOD AYSC sets the modulation type to Single Carrier.

SENSe:WLAN:AY:MOD? may return AYSC, which means the modulation type is Single Carrier.

## [SENSe]:WLAN:AY:Region

Set or query EDMG or PreEDMG region in the Settings control panel when 11ay 4.32 GHz BW is selected.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

Sense commands

## Syntax

[SENSe]:WLAN:AY:Region { EDMG | PRE1edmg | PRE2edmg }

[SENSe]:WLAN:AY:Region?

## Arguments

Select one of the below arguments:

Argument	Description
EDMG	Sets the region to EDMG for all open displays.
PRE1edmg	Sets the region to PRE1edmg for all open displays.
PRE2edmg	Sets the region to PRE2edmg for all open displays.

## Returns

See Arguments.

## Examples

SENSe:WLAN:AY:REG EDMG sets the region to EDMG for all open displays when 11ay 4.32 GHz BW is selected.

`SENSe:WLAN:AY:REG?` may return `PRE1`, which indicates that the region is set to `PRE1edmg` for all open displays when 11ay 4.32 GHz BW is selected.

## **[SENSe]:WLAN:AY:SUMMary:CLEar:RESults (No Query Form)**

Clears all Scalar results in the Summary display.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:SUMMary:CLEar:RESults
```

### **Arguments**

None

### **Examples**

`SENSe:WLAN:AY:SUMM:CLE:RES` clears all Scalar results.

## **[SENSe]:WLAN:AY:SUMMary:LIMIt:EVM**

Set or query the EVM Limit value (`arg1`), which is located under the limit tab in the Summary control panel, and the EVM Limit status (`arg2`) as disabled or enabled.

### **Conditions**

Measurement view: 802.11ay Analysis Summary

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:AY:SUMMary:LIMIt:EVM <arg1, arg2>
```

```
[SENSe]:WLAN:AY:SUMMary:LIMIt:EVM?
```

### **Arguments**

Number of arguments is two: `<arg1, arg2>`. Units: dB or %.

`<arg1>`:

OFF or 0 disables EVM Limit.

ON or 1 enables EVM Limit.

`<arg2>`:

Sets the limit EVM limit value to show pass/fail statuses for the EVM measurement result.

You must select the units (dB or %) in the EVM tab.

## Returns

0 or 1, <NRf>

## Examples

`SENSe:WLAN802_11ad:SUMM:LIMI:EVM 1,-30` where <arg1> is 1, which indicates that Limit is enabled for the EVM Limit comparison and <arg2> is -30, which sets the EVM Limit value at -30%.

`SENSe:WLAN802_11ad:SUMM:LIMI:EVM?` may return 0, which indicates that Limit is disabled for EVM Limit comparison and sets the EVM Limit value at -30 dB.

## [SENSe]:WLAN:AY:SUMMARY:LIMIT:FREQUENCY:ERROR

Set or query the Frequency Error Limit value (arg1), which is located under the limit tab in the Summary control panel, and the Frequency Error Limit status (arg2) as disabled or enabled.

## Conditions

Measurement view: 802.11ay Analysis Summary

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:AY:SUMMARY:LIMIT:FREQUENCY:ERROR <arg1, arg2>`

`[SENSe]:WLAN:AY:SUMMARY:LIMIT:FREQUENCY:ERROR?`

## Arguments

Number of arguments is two: <arg1, arg2>. Units: ppm.

<arg1>:

OFF or 0 turns off Frequency Error Limit.

ON or 1 turns on Frequency Error Limit.

<arg2>:

Specifies show pass/fail statuses for the Frequency Error Limit value.

## Returns

0 or 1, <NRf>

## Examples

`SENSe:WLAN:AY:SUMM:LIMI:FREQ:ERRO 1,20` where <arg1> is 1, which indicates that Limit is enabled for Frequency Error Limit comparison, and <arg2> is 20, which sets the Frequency Error Limit value at 20 ppm.

`SENSe:WLAN:AY:SUMM:LIMI:FREQ:ERRO?` may return 1, 20, where 1 indicates that Limit is enabled for Frequency Error Limit comparison and 20 indicates that the frequency error limit value is 20 ppm.

## **[SENSe]:WLAN:BURSt**

Specifies the burst to measure when multiple bursts are present in an acquisition, in the WLAN analysis parameters. The index number of the first burst is 1, second burst is 2, etc.

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:BURSt <NR1>
```

```
[SENSe]:WLAN:BURSt?
```

### **Arguments**

Integer that specifies which burst to measure when multiple bursts are present in an acquisition.

### **Examples**

`SENSe:WLAN:BURST 3` sets the burst to measure in the presence of multiple bursts to the third burst.

## **[SENSe]:WLAN:CHANnel[:BANDwidth]:BWIDth]**

Specifies or queries a nominal channel bandwidth to use for the WLAN measurements.

To automatically set the subcarrier spacing for the WLAN measurements, use the command

```
[SENSe]:WLAN:SCARriers:SPACing:AUTO.
```

### **Conditions**

Measurement view: WLAN

This command requires WLAN Measurements

### **Group**

Sense commands

### **Syntax**

```
[SENSe]:WLAN:CHANnel[:BANDwidth]:BWIDth {BW5|BW10|BW20|BW40|BW80|BW160}
```

```
[SENSe]:WLAN:CHANnel[:BANDwidth]:BWIDth?
```

## Arguments

BW5, BW10, BW20, BW40, BW80, BW160 specify a nominal channel bandwidth to use for the WLAN measurements.

## Returns

Query returns one of the argument values indicating the selected nominal channel BW.

## Examples

`SENSE:WLAN:CHANNEL:BANDWIDTH BW20` sets the nominal channel bandwidth to use to 20 MHz.

## [SENSE]:WLAN:EQUALIZER:TRAINING

Specifies the method to use for estimating channel frequency response and equalization for WLAN measurements (either PREAMBLE or PDATA). This control can be used to diagnose changes in frequency response over the signal packet.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSE]:WLAN:EQUALIZER:TRAINING {PREAMBLE|PDATA}
```

```
[SENSE]:WLAN:EQUALIZER:TRAINING?
```

## Arguments

**PREAMBLE** The instrument uses only the Preamble to estimate channel frequency response. This response is then used to equalize the entire signal packet.

**PDATA** Preamble plus data. The instrument makes an initial channel frequency response estimate from the Preamble. Then it estimates the channel response for each data symbol using the decoded data content to derive equalization for each symbol individually. This allows compensation for time-varying channel response over the packet.

## Examples

`SENSE:WLAN:EQUALIZER:TRAINING PREAMBLE` sets the method used to estimate channel frequency response and equalization to Preamble.

## [SENSE]:WLAN:FFT:LENGTH? (Query Only)

Returns the fixed FFT length for the WLAN measurements.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:FFT:LENGth?
```

## Returns

Integer that represents the fixed FFT length for the WLAN measurements.

## Examples

`SENSe:WLAN:FFT:LENGTH?` might return the fixed FFT length as 64.

## [SENSe]:WLAN:FFT:SRATE? (Query Only)

Returns the FFT sample rate for the WLAN measurements, in samples per second, based on the bandwidth or subcarrier setting.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:FFT:SRATE?
```

## Returns

Floating point number that represents the FFT sample rate in MS/s.

## Examples

`SENSe:WLAN:FFT:SRATE?` might return 20.000e+6, which represents an FFT sample rate of 20.000 MS/s.

## [SENSe]:WLAN:FREQuency:ERRor

Sets or queries the frequency error in the WLAN analysis parameters. The entered value is used by the analysis as a fixed frequency offset. This is useful when

the exact frequency offset of the signal is known. To set the frequency error automatically, use the command

```
[SENSe]:WLAN:FREQuency:ERRor:AUTO.
```

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:FREQUENCY:ERRor <Nrf>
```

```
[SENSe]:WLAN:FREQUENCY:ERRor?
```

## Arguments

Floating point number that represents the fixed frequency offset.

## Examples

`SENSe:WLAN:FREQUENCY:ERROR 0.02` sets the frequency error to 0.02.

## [SENSe]:WLAN:FREQUENCY:ERRor:AUTO

Turns on the automatic determination of the frequency error in the WLAN analysis parameters. To enter a value manually when Auto is turned off, use the command `[SENSe]:WLAN:FREQUENCY:ERRor`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:FREQUENCY:ERRor:AUTO {0|1|OFF|ON}
```

```
[SENSe]:WLAN:FREQUENCY:ERRor:AUTO?
```

## Arguments

OFF or 0 sets the frequency error manually, using the `[SENSe]:WLAN:FREQUENCY:ERRor` command.

ON or 1 sets the frequency error automatically.

## Examples

`SENSe:WLAN:FREQUENCY:ERROR:AUTO ON` Turns on automatic determination of the frequency error.

## [SENSe]:WLAN:GUARd:INTerval

Sets or queries the Guard Interval to use in the input signal in the WLAN measurements.

## Conditions

Measurement view: WLAN



---

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:GUARd:INTerval {AUTO|FOURth|EIGHth}
```

```
[SENSe]:WLAN:GUARd:INTerval?
```

## Arguments

AUTO Auto from SIG uses the Guard Interval value extracted from the signal

FOURth specifies the guard interval as 1/4.

EIGHth specifies the guard interval as 1/8.

## Examples

SENSe:WLAN:GUARd:INTERVAL FOUR sets the guard interval to 1/4.

## [SENSe]:WLAN:MODulation:DETECT:QAM1024

Sets or queries whether or not 1024QAM detection is enabled.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:MODulation:DETECT:QAM1024 { 0 | 1 }
```

```
[SENSe]:WLAN:MODulation:DETECT:QAM1024?
```

## Arguments

0 means detection is disabled.

1 means detection is enabled.

## Returns

See Arguments

## Examples

SENSe:WLAN:MODulation:MANual DSSS1 specifies to use DSSS1 fixed modulation regardless of the signal content.

## [SENSe]:WLAN:MODulation:MANual

Specifies to use a fixed modulation type, regardless of the signal content, in the WLAN analysis parameters.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:MODulation:MANual {BPSK |QPSK |QAM16 |QAM64|QAM256 |QAM1024 |DSSS1 |  
DSSS2 |CCK5 |CCK11}
```

```
[SENSe]:WLAN:MODulation:MANual?
```

### Arguments

BPSK |QPSK |QAM16 |QAM64 |QAM256 |QAM1024| DSSS1 |DSSS2|CCK5 |CCK11 are fixed modulation types.

### Examples

SENSe:WLAN:MODulation:MANual DSSS1 specifies to use DSSS1 fixed modulation regardless of the signal content.

## [SENSe]:WLAN:MODulation:SOURce

Specifies the method of data symbol modulation identification in the WLAN analysis parameters (either Auto Detect, Manual, or Auto from SIG).

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:MODulation:SOURce [AUTO|MANual|SIG]
```

```
[SENSe]:WLAN:MODulation:SOURce?
```

### Arguments

AUTO Auto Detect estimates the modulation from the data symbol IQ content.

MANual Manual allows specifying a fixed modulation type regardless of the signal content, using the command

```
[SENSe]:WLAN:MODulation:MANual.
```

SIG Auto from SIG sets the modulation as indicated by the embedded SIG preamble symbol format data.

## Examples

`SENSe:WLAN:MODULATION:SOURCE MANUAL` enables you to specify a fixed modulation type regardless of the signal content, using the command `[SENSe]:WLAN:MODulation:DETECT:QAM1024`

## [SENSe]:WLAN:PILOt:TRACking:AMPLitude:STATe

Specifies to use the pilot subcarrier to correct amplitude variations over the packet for WLAN measurements. Note: the default setting is 0 (false).

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:PILOt:TRACking:AMPLitude:STATe {0|1|OFF|ON}
```

```
[SENSe]:WLAN:PILOt:TRACking:AMPLitude:STATe?
```

### Arguments

ON or 1 specifies to use the Amplitude pilot subcarrier to correct amplitude variation over the packets.

OFF or 0 specifies that the Amplitude pilot subcarrier is off.

## Examples

`SENSe:WLAN:PILOt:TRACking:AMPLITUDE:STATE ON` specifies to use the pilot subcarrier to correct variations.

## [SENSe]:WLAN:PILOt:TRACking:PHASe:STATe

Specifies to use the pilot subcarrier to correct phase variations over the packet, in WLAN measurements. Note: the default setting is 1 (true).

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

```
[SENSe]:WLAN:PILOt:TRACking:PHASe:STATe {0|1|OFF|ON}
```

```
[SENSe]:WLAN:PILOt:TRACking:PHASe:STATe?
```

## Arguments

ON or 1 specifies to use the pilot subcarrier to correct phase variations over the packet.

OFF or 0 specifies that the pilot subcarrier is off.

## Examples

`SENSe:WLAN:PILOT:TRACKING:PHASE:STATE ON` specifies to use the pilot subcarrier to correct phase variations over the packet.

## [SENSe]:WLAN:PILOt:TRACkIng:TIMIng:STATe

Specifies to use the pilot subcarrier to correct timing variations over the packet, in WLAN measurements. Note: the default setting is 0 (false).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:PILOt:TRACkIng:TIMIng:STATe {0|1|OFF|ON}
```

```
[SENSe]:WLAN:PILOt:TRACkIng:TIMIng:STATe?
```

## Arguments

ON or 1 specifies to use the pilot subcarrier to correct timing variations over the packet.

OFF or 0 specifies that the pilot subcarrier is off.

## Examples

`SENSe:WLAN:PILOT:TRACKING:TIMING:STATE ON` specifies to use the pilot subcarrier to correct timing variations over the packet.

## [SENSe]:WLAN:RADix

Specifies or queries the format for displaying the symbol values in the WLAN setting preferences (either binary or HEX).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:RADix {BINary|HEX}
```

```
[SENSe]:WLAN:RADix?
```

## Arguments

**BINary** specifies to display the symbol values in binary format.

**HEX** specifies to display the symbol values in HEX format.

## Examples

`SENSe:WLAN:RADIX HEX` specifies to display the symbol values in HEX format.

## [SENSe]:WLAN:SCARriers

Sets or queries which subcarriers to display in the WLAN view.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers {ALL|PILOts|DATA|RANGeE|SINGLe}
```

```
[SENSe]:WLAN:SCARriers?
```

## Arguments

**ALL** specifies that all subcarriers will be displayed.

**PILOt** specifies that only the pilot subcarriers will be displayed.

**DATA** specifies that only the data subcarriers will be displayed.

**RANGe** specifies that the subcarriers within a specified range will be displayed.

**SINGLe** specifies that only a single subcarrier will be displayed.

## Examples

`SENSe:WLAN:SCARRIERS ALL` sets the WLAN view to display all subcarriers.

## [SENSe]:WLAN:SCARriers:RANGe:START

Sets or queries the start value of the subcarrier range to be displayed in the WLAN view.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers:RANGe:START <NR1>
```

```
[SENSe]:WLAN:SCARriers:RANGe:START?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SCARRIERS:RANGE:START 20` sets the subcarrier start range to 20.

## [SENSe]:WLAN:SCARriers:RANGe:STOP

Sets or queries the stop value of the subcarrier range to be displayed in the WLAN view.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers:RANGe:STOP <NR1>
```

```
[SENSe]:WLAN:SCARriers:RANGe:STOP?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SCARRIERS:RANGE:STOP 25` sets the subcarrier stop range to 25.

## [SENSe]:WLAN:SCARriers:SINGLE:INDEX

Specifies or queries the single subcarrier to display in the WLAN view.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers:SINGLE:INDEX <NR1>
```

```
[SENSe]:WLAN:SCARriers:SINGLE:INDEX?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SCARRIERS:SINGLE:INDEX -10` sets the single subcarrier for display to -10.

## [SENSe]:WLAN:SCARriers:SPACing

Sets or queries the subcarrier spacing in the WLAN measurements.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers:SPACing <NRf>
```

```
[SENSe]:WLAN:SCARriers:SPACing?
```

## Arguments

NR3 is a floating point number that specifies the subcarrier spacing.

## Returns

`SENSe:WLAN:SCARRIERS:SPACING 312.5E+3` sets the subcarrier spacing to 312.5 kHz.

## [SENSe]:WLAN:SCARriers:SPACing:AUTO

When turned on, automatically sets the subcarrier spacing in the WLAN measurements.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SCARriers:SPACing:AUTO {0|1|OFF|ON}
```

```
[SENSe]:WLAN:SCARriers:SPACing:AUTO?
```

## Arguments

OFF or 0 turns the automatic subcarrier spacing off.

ON or 1 turns the automatic subcarrier spacing on.

## Examples

`SENSe:WLAN:SCARriers:SPACing:AUTO ON` turns the automatic subcarrier spacing on.

## [SENSe]:WLAN:SIGNal:STANdard:TYPE

Sets or queries the WLAN signal standard.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

Additionally, the N argument requires Option 24, and the AC argument requires Option 25.

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SIGNal:STANdard:TYPE {A|B|G|J|P|N|AC}
```

```
[SENSe]:WLAN:SIGNal:STANdard:TYPE?
```

## Arguments

A sets the WLAN signal standard to 802.11a.

B sets the WLAN signal standard to 802.11b.

G sets the WLAN signal standard to 802.11g.

J sets the WLAN signal standard to 802.11j.

P sets the WLAN signal standard to 802.11p.

N sets the WLAN signal standard to 802.11n. Requires Option 24 in addition to Option 23.



AC sets the WLAN signal standard to 802.11ac. Requires Option 25 in addition to Option 23.

## Examples

`SENSe:WLAN:SIGNAL:STANDARD:TYPE G` sets the WLAN signal standard to 802.11g.

## [SENSe]:WLAN:SubCARrier:DERotation

Turning on this state allows some displays to show subcarriers with or without the Gamma subcarrier phase rotation removed. Gamma phase rotation is applied to 802.11n and 802.11ac subcarriers in defined subranges depending on the Channel Bandwidth selected (only for Channel Bandwidths of 40 MHz and above).

When derotation is turned off, the rotation is not removed, which provides a direct view of the physical modulation on the channel. With derotation turned on, the rotation is removed, allowing easier decoding of the underlying data content.

Only the Constellation and Symbol Table results are affected by this control.



**Note:** This control is only shown for 802.11n and 802.11ac standard selections for channel bandwidths greater than or equal to 40 MHz. It is not present for all others.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:SubCARrier:DERotation {0|1|OFF|ON}`

`[SENSe]:WLAN:SubCARrier:DERotation?`

## Arguments

OFF or 0 turns the subcarrier derotation off.

ON or 1 turns the subcarrier derotation on.

## Examples

`SENSe:WLAN:SUBCARRIER:DEROTATION ON` turns the subcarrier derotation on.

## [SENSe]:WLAN:SUMMARY:CLEar:RESults (No Query Form)

Clears the results in the WLAN Summary measurement.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SUMMARY:CLEAR:RESULTS
```

## Arguments

None

## Examples

`SENSe:WLAN:SUMMARY:CLEAR:RESULTS` clears the results in the measurement.

## [SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT

Specifies the maximum number of bursts to average for the WLAN Summary EVM display. The burst average function must be enabled using the command `[SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT:ENABLE`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT <NR1>
```

```
[SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT 12` sets the maximum number of bursts to average to 12.

## [SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT:ENABLE

Enables the burst average function in the WLAN Summary EVM settings. To specify the maximum number of bursts to average, use the command `[SENSe]:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT`.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SUMMary:EVM:RMS:AVERage:COUNT:ENABLE{0|1|OFF|ON}
```

```
[SENSe]:WLAN:SUMMary:EVM:RMS:AVERage:COUNT:ENABLE?
```

## Arguments

ON or 1 turns the burst average function on.

OFF or 0 turns the burst average function off.

## Examples

`SENSe:WLAN:SUMMARY:EVM:RMS:AVERAGE:COUNT:ENABLE ON` turns the burst average function on.

## [SENSe]:WLAN:SWAP:IQ

Specifies to swap the I and Q components of a signal in the WLAN analysis parameters. Doing so compensates the input signal for spectral inversion.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SWAP:IQ {0|1|OFF|ON}
```

```
[SENSe]:WLAN:SWAP:IQ?
```

## Arguments

OFF or 0 turns the Swap I and Q function off.

ON or 1 turns the Swap I and Q function on.

## Examples

`SENSe:WLAN:SWAP:IQ ON` swaps the I and Q components of the signal.

## [SENSe]:WLAN:SYMBOLS|CHIPS

Specifies or queries which symbols or chips are displayed in the WLAN Symbol Table.

Use "CHIPS" for 802.11b and "SYMBOLS" for all other standards. The DAT, PRAM and HEAD enumerations apply to 802.11b only.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SYMBOLs|CHIPs {ALL|RANGe|SINGLe|DAT|PRAM|HEAD}
```

```
[SENSe]:WLAN:SYMBOLs|CHIPs?
```

## Arguments

ALL specifies that all symbols or chips will be displayed.

RANGe specifies that a range of symbols or chips will be displayed.

SINGLe specifies that single symbols or chips will be displayed.

DAT specifies that the Data symbols or chips will be displayed. Applies to 802.11b only.

PRAM specifies that Preamble symbols or chips will be displayed. Applies to 802.11b only.

HEAD specifies that Header symbols or chips will be displayed. Applies to 802.11b only.

## Examples

SENSe:WLAN:SYMBOLS ALL specifies that all symbols or chips will be displayed in the WLAN Symbol Table.

## [SENSe]:WLAN:SYMBOLs|CHIPs:MAX

Sets or queries the value for the maximum number of symbols/chips to analyze for the WLAN Symbol Table.

Use "CHIPs" for 802.11b and "SYMBOLs" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SYMBOLs|CHIPs:MAX <NR1>
```

```
[SENSe]:WLAN:SYMBOLs|CHIPs:MAX?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SYMBOLS:MAX 50` specifies a maximum value of 50 symbols to analyze for the WLAN Symbol Table.

## [SENSe]:WLAN:SYMBOLS|CHIPS:MAX:STATE

Turns on or off the “maximum data symbols to analyze” feature for the WLAN Symbol Table. To specify the maximum number of symbols/chips to analyze, use the command.

Use "CHIPS" for 802.11b and "SYMBOLS" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SYMBOLS|CHIPS:MAX:STATE {0|1|OFF|ON}
```

```
[SENSe]:WLAN:SYMBOLS|CHIPS:MAX:STATE?
```

## Arguments

ON or 1 turns on the “maximum data symbols to analyze” feature.

OFF or 0 disables the “maximum data symbols to analyze” feature.

## Examples

`SENSe:WLAN:SYMBOLS:MAX:STATE ON` turns on “maximum data symbols to analyze” feature for the WLAN Symbol Table.

## [SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:COUNT

Sets or queries the number of symbols or chips to display when displaying a range results in the WLAN Symbol Table.

Use "CHIPS" for 802.11b and "SYMBOLS" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:COUNT <NR1>
```

```
[SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:COUNT?
```

## Arguments

Integer that represents the number of symbols/chips to display

## Examples

`SENSe:WLAN:SYMBOLS:RANGE:COUNT 4` sets the number of symbols or chips to display to 4 in the WLAN Symbol Table.

## [SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:START

Sets or queries the symbols/chips start value when displaying a range of results in the WLAN Symbol Table.

Use "CHIPS" for 802.11b and "SYMBOLS" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:START <NR1>`

`[SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:START?`

## Arguments

Integer

## Examples

`SENSe:WLAN:SYMBOLS:RANGE:START 40` sets the start value of the range to 40.

## [SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:STOP

Sets or queries the symbols or chips stop value when displaying a range of results in the WLAN Symbol Table.

Use "CHIPS" for 802.11b and "SYMBOLS" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

`[SENSe]:WLAN:SYMBOLS|CHIPS:RANGE:STOP <NR1>`

```
[SENSe]:WLAN:SYMBOLs|CHIPs:RANGe:STOP?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SYMBOLS:RANGE:STOP 60` sets the stop value of the range to 60.

## [SENSe]:WLAN:SYMBOLs|CHIPs:SINGLE:INDEX

Sets or queries the symbol or chip to use when displaying results from a single symbol in the WLAN Symbol Table.

Use "CHIPs" for 802.11b and "SYMBOLs" for all other standards.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:SYMBOLs|CHIPs:SINGLE:INDEX <NR1>
```

```
[SENSe]:WLAN:SYMBOLs|CHIPs:SINGLE:INDEX?
```

## Arguments

Integer

## Examples

`SENSe:WLAN:SYMBOLs|CHIPs:SINGLE:INDEX 26` specifies to display symbol 26.

## [SENSe]:WLAN:UNIT:FREQUENCY

Specifies or queries whether the displayed frequency units are frequency (Hz) or subcarrier in the WLAN settings.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Sense commands

## Syntax

```
[SENSe]:WLAN:UNIT:FREQUENCY {FREQUENCY|SCARrier}
```

[SENSe]:WLAN:UNIT:FREQuency?

### Arguments

FREQuency specifies that the displayed frequency units will be Hz.

SCARrier specifies that the displayed frequency units will be subcarrier.

### Examples

SENSe:WLAN:UNIT:FREQUENCY FREQ specifies that the frequency units in the display will be Hz.

## [SENSe]:WLAN:UNIT:TIME

Specifies or queries whether the displayed time units are seconds, symbols or chips in the WLAN settings. Chips is only a valid option for 802.11b, and Symbols is valid for all other standards.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Sense commands

### Syntax

[SENSe]:WLAN:UNIT:TIME {SECOnds|SYMBols|CHIPs}

[SENSe]:WLAN:UNIT:TIME?

### Arguments

SECOnds specifies that the displayed time units are to be seconds

SYMBols specifies that the displayed time units are to be symbols

CHIPs specifies that the displayed time units are to be chips (valid only for 802.11b)

### Examples

SENSe:WLAN:UNIT:TIME SYMBOLS specifies that the displayed time units are to be symbols.

## SOURce:TXGain:POWer

Sets or queries the output power level of the tracking generator.

### Conditions

Measurement view: Transmission Gain

### Group

Source commands



## Syntax

```
SOURce:TXGain:POWer <value>
```

## Arguments

<value>::=<NRf> is the tracking generator output power.

Range: -3 dBm to -43 dBm.

## Returns

See Arguments.

## Examples

```
SOURCE:TXGAIN:POWER-10 sets the tracking generator output power to -10 dBm.
```

## \*SRE

Sets or queries the value of the Service Request Enable Register (SRER).

Refer to Section 3, Status and Events, for the register information.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

```
*SRE <value>
```

```
*SRE?
```

## Related Commands

```
*CLS, *ESE, *ESR?, *STB?
```

## Arguments

<value>::<NR1> is a value in the range from 0 to 255.

The binary bits of the SRER are set according to this value.

Using an out-of-range value causes an execution error.

## Examples

```
*SRE48 sets binary 00110000 in the SRER's bits.
```

```
*SRE? might return 32, indicating that binary value 00100000 has been set in the SRER's bits.
```

## Status commands

### STATUS:ACPower:EVENTs? (Query Only)

Returns the current events and status conditions for the Channel power and ACPR measurement.

#### Conditions

Measurement views: Channel power and ACPR

#### Group

Status commands

#### Syntax

```
STATUS:ACPower:EVENTs?
```

#### Arguments

None

#### Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

#### Examples

STATUS:ACPOWER:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

### STATUS:{AM|FM|PM}:EVENTs? (Query Only)

Returns the current events and status conditions for the AM/FM/PM measurement.

#### Conditions

Measurement views: AM, FM, PM

#### Group

Status commands

#### Syntax

```
STATUS:{AM|FM|PM}:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:AM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually in the AM measurement.

## STATus:AUDio:SPECTrum:EVENTs? (Query Only)

Returns the current events and status conditions for the audio measurement.

## Conditions

Measurement views: Audio Spectrum

## Group

Status commands

## Syntax

```
STATus:AUDio:SPECTrum:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where:

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:AUDIO:SPECTRUM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:AUDio:SUMMary:EVENTs? (Query Only)

Returns the current events and status conditions for the audio measurement.

### Conditions

Measurement views: Audio Summary

### Group

Status commands

### Syntax

STATus:AUDio:SUMMary:EVENTs?

### Arguments

None

### Returns

<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc> [:<einfo>]" }

Where:

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event.

If there is no error, the response is 0, "No events to report".

### Examples

STATus:AUDio:SUMMary:EVENTs? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:AVTime[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions for the Amplitude versus Time measurement.

### Conditions

Measurement views: Amplitude versus Time

### Group

Status commands

### Syntax

STATus:AVTime[:MEASview<y>]:EVENTs?

## Arguments

The parameter  $\langle y \rangle = 1$  to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default  $y$  is 1

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

$\langle \text{ecode} \rangle ::= \langle \text{NR1} \rangle$  is the error/event code (-32768 to 32767).

$\langle \text{edesc} \rangle ::= \langle \text{string} \rangle$  is the description on the error/event.

$\langle \text{einfo} \rangle ::= \langle \text{string} \rangle$  is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:AVTIME:MEASVIEW1:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually for Amplitude versus Time Measurement View 1.

## STATUS:BIBEmissions:EVENTs? (Query Only)

Returns the current events and status conditions for the Bluetooth InBand Emission measurement.

## Conditions

Measurement views: Bluetooth InBand Emission

## Group

Status commands

## Syntax

```
STATUS:BIBEmissions:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[,<einfo>]" {, "<edesc>[,<einfo>]" }
```

Where:

$\langle \text{ecode} \rangle ::= \langle \text{NR1} \rangle$  is the error/event code (-32768 to 32767).

$\langle \text{edesc} \rangle ::= \langle \text{string} \rangle$  is the description on the error/event.

$\langle \text{einfo} \rangle ::= \langle \text{string} \rangle$  is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:BIBEMISSIONS:EVENTS? might return 0 indicating no events to report.

## STATUS:BLUetooth:CONStE:EVENTs? (Query Only)

Returns the current events and status conditions for the Bluetooth Constellation measurement.

### Conditions

Measurement views: Bluetooth Constellation

### Group

Status commands

### Syntax

```
STATUS:BLUetooth:CONStE:EVENTs?
```

### Arguments

None

### Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>,"<edesc>[<einfo>]" }
```

Where

<ecode> ::= <NR1> is the error/event code (-32768 to 32767).

<edesc> ::= <string> is the description of the error/event.

<einfo> ::= <string> is the additional information of the error/event.

### Examples

STATUS:BLUetooth:CONStE:EVENTs? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:BLUetooth:EDIAGram:EVENTs? (Query Only)

Returns the current events and status conditions for the eye diagram measurement.

### Conditions

Measurement views: Bluetooth Eye Diagram

### Group

Status commands

### Syntax

```
STATUS:BLUetooth:EDIAGram:EVENTs?
```

### Arguments

None

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[<einfo>]" }
```

Where

<ecode> ::= <NR1> is the error/event code (-32768 to 32767).

<edesc> ::= <string> is the description of the error/event.

<einfo> ::= <string> is the additional information of the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:BLUETOOTH:EDIAGRAM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:BLUETOOTH:FDVTIME:EVENTS? (Query Only)

Returns the current events and status conditions for the Frequency Deviation versus Time measurement.

## Conditions

Measurement views: Frequency Deviation vs. Time

## Group

Status commands

## Syntax

```
STATUS:BLUETOOTH:FDVTIME:EVENTS?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[,<einfo>]" {, "<edesc>[,<einfo>]" }
```

Where:

<ecode> ::= <NR1> is the error/event code (-32768 to 32767).

<edesc> ::= <string> is the description on the error/event.

<einfo> ::= <string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:BLUETOOTH:FDVTIME:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:BLUETOOTH:SUMMARY:EVENTS (Query Only)

Returns the current events and Status conditions for the Bluetooth Summary measurements.

### Conditions

Measurement view: Bluetooth Summary display

### Group

Status commands

### Syntax

STATUS:BLUETOOTH:SUMMARY:EVENTS?

### Arguments

None

### Returns

`<ecode>,"<edesc>[<info>"]{,<ecode>,"<edesc>[:<info>"]}` where

`<ecode>::` <NR1> is the error/event code (-32768 to 32767).

`<edesc>::` <String> is the description of the error/event.

`<info>::` <String> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

### Examples

STATUS:BLUETOOTH:SUMMARY:EVENTS might return 12026, "Acq Sampling params: manual control".

## STATUS:CCDF:EVENTS? (Query Only)

Returns the current events and status conditions for the CCDF measurement.

### Conditions

Measurement views: CCDF

### Group

Status commands

### Syntax

STATUS:CCDF:EVENTS?

### Arguments

None

### Returns

`<ecode>,"<edesc>[<info>"]{,<ecode>,"<edesc>[:<info>"]}`



## Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:CCDF:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:CONSte[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the constellation measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Status commands

## Syntax

```
STATus:CONSte[:MEASview<y>]:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

## Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:CONSte:MEASview2:EVENTs? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually for measview 2 of constellation.

## **STATus:DIQVtime[:MEASview<y>]:EVENTs? (Query only)**

Returns the current events and status conditions for the Demod I&Q versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### **Conditions**

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### **Group**

Status commands

### **Syntax**

```
STATus:DIQVtime[:MEASview<y>]:EVENTs?
```

### **Arguments**

None

### **Returns**

```
<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}
```

Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error or event.

<einfo>::<string> is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

### **Examples**

STATus:DIQVtime:MEASview1:EVENTs? might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually for measurement view 1.

## **STATus:DPX:EVENTs? (Query Only)**

Returns the current events and status conditions for the DPX, Spectrum measurement.

### **Conditions**

Measurement views: DPX, Spectrum

### **Group**

Status commands

### **Syntax**

```
STATus:DPX:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:DPX:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:EDiagram[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the eye diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Status commands

## Syntax

```
STATus:EDIagram[:MEASview<y>]:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error or event.

<einfo>::<string> is the additional information on the error or event.

If there is no error, the response is 0, "No events to report".

## Examples

`STATus:EDIagram:MEASview2:EVENTs?` might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:EMC:EVENTs? (Query Only)

Returns the current events and status conditions for the EMC display.

### Conditions

Measurement view: EMC - EMI Display

### Group

Status commands

### Syntax

`STATus:EMC:EVENTs??`

### Arguments

None

### Returns

`<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }`

Where

`<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

### Examples

`STATUS:EMC:EVENTS??` might return 11037, "Ext corrections enabled for traces."

## STATus:EVM[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the EVM versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod EVM versus Time).

## Group

Status commands

## Syntax

```
STATus:EVM[:MEASview<y>]:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error or event.

<einfo>::<string> is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:EVM:MEASview2:EVENTs? might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:FDVTime[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the Frequency deviation versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

## Group

Status commands

## Syntax

```
STATus:FDVTime:MEASview<y>:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

### Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error or event.

<einfo>::<string> is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

### Examples

STATUS:FDVTIME:MEASview2:EVENTS? might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:{FSETtling|PSETtling}:EVENTs? (Query Only)

Returns the current events and status conditions for the specified settling time measurement.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Status commands

### Syntax

```
STATUS:{FSETtling|PSETtling}:EVENTs?
```

### Arguments

None

### Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

### Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

### Examples

STATUS:{FSETTLING|PSETTLING}:EVENTS? might return 2048,"Done saving - restart with Stop, then Run", indicating that the acquisition has been completed.

## STATUS:FVTime[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions for the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

## Group

Status commands

## Syntax

```
STATus:FVTime[:MEASview<y>]:EVENTs?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:FVTIME:MEASview1:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:IQVTime[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions for the RF I&Q versus Time measurement.

## Conditions

Measurement views: RF I&Q versus Time

## Group

Status commands

## Syntax

```
STATus:IQVTime[:MEASview<y>]:EVENTs?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:IQVTIME:MEASview1:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:LTE:ACLR:EVENTs? (Query Only)

Returns the current events and status conditions for the LTE ACLR measurement.

### Conditions

Measurement view: LTE ACLR

### Group

Status commands

### Syntax

`STATus:LTE:ACLR:EVENTs?`

### Returns

`<einfo>,"<edesc>[<einfo>]"{"<einfo>,"<edesc>[:<einfo>]"}`

Where

`<einfo>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description of the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

`STATUS:LTE:ACLR:EVENTS?` might return 12006, "Not aligned",2008,"Not calibrated",12007,"unaligned data",12009,"Data from uncalibrated instrument".

## STATus:LTE:CHSPectrum:EVENTs? (Query Only)

Returns the current events and status conditions for the LTE Channel Spectrum measurement.

### Conditions

Measurement view: LTE Channel Spectrum

### Group

Status commands

### Syntax

`STATus:LTE:CHSPectrum:EVENTs?`



## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATUS:LTE:CHSPECTRUM:EVENTS? might return 12006,"Not aligned",2008,"Not calibrated".

## STATUS:LTE:CONStE:EVENTs? (Query Only)

Returns the current events and status condition for the LTE Constellation measurement.

### Conditions

Measurement view: LTE Constellation

### Group

Status commands

### Syntax

```
STATUS:LTE:CONStE:EVENTs?
```

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATUS:LTE:CONStE:EVENTS? might return 12006,"Not aligned",2008,"Not calibrated",11090,"Recovery done on PSS/SSS on the center 62, "106 - AnaSynchronization Sequence not found".

## STATUS:LTE:PVTime:EVENTs? (Query Only)

Returns the current events and status conditions for the LTE Power vs Time measurement.

### Conditions

Measurement view: LTE Power vs Time

## Group

Status commands

## Syntax

```
STATUS:LTE:PVTIME:EVENTS?
```

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

```
STATUS:LTE:PVTIME:EVENTS? might return 12006,"Not aligned",2008,"Not calibrated".
```

## STATUS:MCPower:EVENTS? (Query Only)

Returns the current events and status conditions for the MCPR (multi-carrier ACPR) measurement.

## Conditions

Measurement views: MCPR

## Group

Status commands

## Syntax

```
STATUS:MCPower:EVENTS?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:MCPOWER:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:MERRor[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the Magnitude error versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Magnitude error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Status commands

## Syntax

`STATus:MERRor[:MEASview<y>]:EVENTs?`

## Arguments

None

## Returns

`<ecode>,"<edesc>[<einfo>]" { ,<ecode>,"<edesc>[:<einfo>]" }`

Where

`<ecode>::<NR1>` is the error or event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error or event.

`<einfo>::<string>` is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:MERRor:MEASview2:EVENTS?` might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OBWidth:EVENTs? (Query Only)

Returns the current events and status conditions for the Occupied Bandwidth (OBW) measurement.

## Conditions

Measurement views: Occupied Bandwidth

## Group

Status commands

## Syntax

STATus:OBwidth:EVENTs?

## Arguments

None

## Returns

<code>, "<edesc> [<einfo>]" {, <code>, "<edesc> [:<einfo>]" }

Where

<code>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:OBWIDTH:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:CONStE:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM Constellation measurement.

## Conditions

Measurement views: OFDM

## Group

Status commands

## Syntax

STATus:OFDM:CONStE:EVENTs?

## Arguments

None

## Returns

<code>, "<edesc> [<einfo>]" {, <code>, "<edesc> [:<einfo>]" }

Where

<code>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:OFDM:CONSTE:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:EVM:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM EVM (Error Vector Magnitude) versus Time measurement.

### Conditions

Measurement views: OFDM

### Group

Status commands

### Syntax

`STATus:OFDM:EVM:EVENTs?`

### Arguments

None

### Returns

`<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }`

Where

`<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:OFDM:EVM:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:MERRor:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM Magnitude versus Time measurement.

### Conditions

Measurement views: OFDM

### Group

Status commands

## Syntax

STATus:OFDM:MERRor:EVENTs?

## Arguments

None

## Returns

<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:OFDM:MERRor:EVENTs? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:PERRor:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM Phase error versus Time measurement.

## Conditions

Measurement views: OFDM

## Group

Status commands

## Syntax

STATus:OFDM:PERRor:EVENTs?

## Arguments

None

## Returns

<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:OFDM:PERROR:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:POWer:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM Power measurement.

### Conditions

Measurement views: OFDM

### Group

Status commands

### Syntax

STATus:OFDM:POWer:EVENTs?

### Arguments

None

### Returns

<ecode>,"<edesc>[<einfo>]" { ,<ecode>,"<edesc>[:<einfo>]" }

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:OFDM:POWER:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OFDM:STABLE:EVENTs? (Query Only)

Returns the current events and status condition for the OFDM Symbol table measurement.

### Conditions

Measurement views: OFDM

### Group

Status commands

### Syntax

STATus:OFDM:STABLE:EVENTs?

## Arguments

None

## Returns

`<ecode>, "<edesc> [<einfo>"] "{, <ecode>, "<edesc>[:<einfo>]" }`

Where

`<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:OFDM:STABLE:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:OPERation:CONDition? (Query Only)

Returns the contents of the Operation Condition Register (OCR).

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

`STATus:OPERation:CONDition?`

## Arguments

None

## Returns

`<NR1>` is a decimal number showing the contents of the OCR.

## Examples

`STATUS:OPERATION:CONDITION?` might return 16, showing that the bits in the OCR have the binary value 000000000010000, which means the analyzer is in measurement.

## STATus:OPERation:ENABLE

Sets or queries the enable mask of the Operation Enable Register (OENR) which allows true conditions in the Operation Event Register to be reported in the summary bit.



## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:OPERATION:ENABLE <bit_value> STATUS:OPERATION:ENABLE?
```

## Arguments

<bit\_value>::<NR1> is the enable mask of the OENR. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the OENR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

STATUS:OPERATION:ENABLE1 enables the ALIGNing bit.

STATUS:OPERATION:ENABLE? might return 1, showing that the bits in the OENR have the binary value 00000000 00000001, which means that the ALIGNing bit is valid.

## STATUS:OPERATION[:EVENT]? (Query Only)

Returns the contents of the Operation Event Register (OEVR). Reading the OEVR clears it.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:OPERATION[:EVENT]?
```

## Arguments

None

## Returns

<NR1> is a decimal number showing the contents of the OEVR.

## Examples

STATUS:OPERATION:EVENT? might return 1, showing that the bits in the OEVR have the binary value 00000000 00000001, which means that the ALIGNing bit is set.

## STATus:OPERation:NTRansition

Sets or queries the negative transition filter value of the Operation Transition Register (OTR).

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

```
STATus:OPERation:NTRansition <bit_value> STATus:OPERation:NTRansition?
```

### Arguments

<bit\_value>::<NR1> is the negative transition filter value. Range: 0 to 65535.

### Returns

<NR1> is a decimal number showing the contents of the OTR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

### Examples

STATUS:OPERATION:NTRANSITION#H0011 sets the negative transition filter value to #H0011.

STATUS:OPERATION:NTRANSITION? might return 17.

## STATus:OPERation:PTRansition

Sets or queries the positive transition filter value of the Operation Transition Register (OTR).

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

```
STATus:OPERation:PTRansition <bit_value> STATus:OPERation:PTRansition?
```

### Arguments

<bit\_value>::<NR1> is the positive transition filter value. Range: 0 to 65535.

### Returns

<NR1> is a decimal number showing the contents of the OTR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

### Examples

STATUS:OPERATION:PTRANSITION0 sets the positive transition filter value to 0.

STATUS:OPERATION:PTRANSITION? might return 0.

## STATus:P25:EDIagram:EVENTs? (Query Only)

Returns the current events and status conditions for the P25 Eye Diagram display.

### Conditions

Measurement view: P25 Eye Diagram

### Group

Status commands

### Syntax

STATus:P25:EDIagram:EVENTs?

### Returns

```
<ecode>,"<edesc>[<einfo>]" {,<ecode>,"<edesc>[:<einfo>]" }
```

Where:

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

### Examples

STATUS:P25:EDIAGRAM:EVENTS? might return 12026,"Acq Sampling Parma's: Manual control", indicating that the sampling parameters are controlled Manually.

## STATus:P25:SUMMARY:EVENTs (Query Only)

Returns the current events and status conditions for the Summary measurement.

### Conditions

Measurement views: P25 Summary

This command requires P25 Measurements.

### Group

Status commands

### Syntax

STATus:P25:SUMMARY:EVENTs

### Arguments

None

## Returns

`<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}`

Where

`<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description of the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

`STATUS:P25:SUMMARY:EVENTS` might return 12026 "Acquisition Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:PERRor[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the Phase error versus Time measurement.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Status commands

## Syntax

`STATus:PERRor[:MEASview<y>]:EVENTs?`

## Arguments

None

## Returns

`<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}`

Where

`<ecode>::<NR1>` is the error or event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error or event.

`<einfo>::<string>` is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

## Examples

`STATus:PERRor:MEASview2:EVENTs?` might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually for measurement view 2.

## STATus:PHVTime[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions for the Phase versus Time measurement.

### Conditions

Measurement views: Phase versus Time

### Group

Status commands

### Syntax

```
STATus:PHVTime[:MEASview<y>]:EVENTs?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

### Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

### Examples

STATus:PHVTIME:MEASview1:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:PNOise:EVENTs? (Query Only)

Returns the current events and status conditions for the phase noise measurement.

### Conditions

Measurement views: Spurious

### Group

Status commands

### Syntax

```
STATus:PNOise:EVENTs?
```

### Arguments

None

## Returns

`<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }`

Where

`<ecode>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<einfo>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:PNOISE:EVENTS?` might return `12026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

## STATus:PRESet (No Query Form)

Presets the SCPI enable registers and transition registers.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

`STATus:PRESet`

### Arguments

None

## Examples

`STATUS:PRESET` presets the SCPI enable registers and transition registers.

## STATus:PULSe:CUMulative:HISTogram[:MEASview<y>]:EVENTs (Query Only)

Returns the current events and status conditions for the Cumulative Histogram display.

### Conditions

Measurement views: Cumulative Histogram with the specified Measurement View Number

### Group

Status commands

### Syntax

`STATus:PULSe:CUMulative:HISTogram[:MEASview<y>]:EVENTs?`

## Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATus:PULSe:CUMulative:HISTogram:MEASview1:EVENTS might return 0,"No events to report."

## STATus:PULSe:CUMulative:STATistics[:MEASview<y>]:EVENTs (Query Only)

Returns the current events and status conditions for the Cumulative Statistics display.

## Conditions

Measurement views: Pulse Cumulative Statistics

## Group

Status commands

## Syntax

```
STATus:PULSe:CUMulative:STATistics[:MEASview<y>]:EVENTs?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

```
<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATus:PULSe:CUMulative:STATistics:MEASview1:EVENTs? might return 0,"No events to report".

## STATus:PULSe:OGRAM[:MEASview<y>]:EVENTs (Query Only)

Returns the current events and status conditions for the Pulse-Ogram display.

### Conditions

Measurement views: Pulse-Ogram

### Group

Status commands

### Syntax

```
STATus:PULSe:OGRAM[:MEASview<y>]:EVENTs?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

### Returns

```
<ecode>,"<edesc>[<einfo>"]{,<ecode>,"<edesc>[:<einfo>"]}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description of the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

### Examples

STATus:PULSe:OGRAM:MEASview1:EVENTs? might return 0,"No events to report".

## STATus:PULSe:RESult:EVENTs? (Query Only)

Returns the current events and status conditions for the pulse table measurement.

### Conditions

Measurement views: Pulse table

### Group

Status commands

### Syntax

```
STATus:PULSe:RESult:EVENTs?
```

### Arguments

None



## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:PULSE:RESULT:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:PULSe:STATistics[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions for the pulse statistics measurement.

### Conditions

Measurement views: Pulse statistics

### Group

Status commands

### Syntax

```
STATus:PULSe:STATistics[:MEASview<y>]:EVENTs?
```

### Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:PULSe:STATistics:MEASview1:EVENTs? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:PULSe:TRACe[:MEASview<y>]:EVENTs

Returns the current events and status conditions for the pulse trace measurement.

## Conditions

Measurement views: Pulse trace with specified Measurement view number

## Group

Status commands

## Syntax

```
STATus:PULSe:TRACe[:MEASview<y>]:EVENTs?
```

## Arguments

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:PULSe:TRACe:MEASVIEW1:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually on pulse trace measurement view 1.

## STATus:QUESTIONable:CALibration:CONDition? (Query Only)

Returns the contents of the questionable calibration condition register.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATus:QUESTIONable:CALibration:CONDition?
```

## Arguments

None

## Returns

<NR1> is a decimal number showing the contents of the questionable calibration condition register.

## Examples

`STATUS:QUESTIONABLE:CALIBRATION:CONDITION?` might return 16384, showing that the bits in the questionable calibration condition register have the binary value 01000000 00000000, which means the Alignment Needed bit is set.

## STATus:QUEStionable:CALibration:ENABle

Sets or queries the enable mask of the questionable calibration enable register which allows true conditions in the questionable calibration event register to be reported in the summary bit.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

`STATus:QUEStionable:CALibration:ENABle <bit_value> STATus:QUEStionable:CALibration:ENABle?`

### Arguments

`<bit_value>::<NR1>` is the enable mask of the questionable calibration enable register. Range: 0 to 65535.

### Returns

`<NR1>` is a decimal number showing the contents of the questionable calibration enable register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

`STATUS:QUESTIONABLE:CALIBRATION:ENABle16384` enables the Alignment Needed bit.

`STATUS:QUESTIONABLE:CALIBRATION:ENABle?` might return 16384, showing that the bits in the questionable calibration enable register have the binary value 01000000 00000000, which means that the Calibration Summary bit is valid.

## STATus:QUEStionable:CALibration[:EVENT]? (Query Only)

Returns the contents of the questionable calibration event register. Reading the register clears it.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

`STATus:QUEStionable:CALibration[:EVENT]?`

## Arguments

None

## Returns

<NR1> is a decimal number showing the contents of the questionable calibration event register.

## Examples

`STATUS:QUESTIONABLE:CALIBRATION:EVENT?` might return 16384, showing that the bits in the questionable calibration event register have the binary value 01000000 00000000, which means that the Calibration Summary bit is set.

## STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION

Sets or queries the negative transition filter value of the questionable calibration transition register.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION <bit_value>  
STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION?
```

## Arguments

<bit\_value>::<NR1> is the negative transition filter value. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the questionable calibration transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

`STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION#H4000` sets the negative transition filter value to #H4000.

`STATUS:QUESTIONABLE:CALIBRATION:NTRANSITION?` might return 16384.

## STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION

Sets or queries the positive transition filter value of the questionable calibration transition register.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```

STATus:QUEStionable:CALibration:PTRansition <bit_value>
STATus:QUEStionable:CALibration:PTRansition?

```

## Arguments

<bit\_value>::<NR1> is the positive transition filter value. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the questionable calibration transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION0 sets the positive transition filter value to 0.

STATUS:QUESTIONABLE:CALIBRATION:PTRANSITION? might return 0.

## STATus:QUEStionable:CONDition? (Query Only)

Returns the contents of the Questionable Condition Register (QCR).

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```

STATus:QUEStionable:CONDition?

```

## Arguments

None

## Returns

<NR1> is a decimal number showing the contents of the QCR.

## Examples

STATUS:QUESTIONABLE:CONDITION? might return 256, showing that the bits in the QCR have the binary value 0000000100000000, which means the Calibration Summary bit is set.

## STATus:QUEStionable:ENABLE

Sets or queries the enable mask of the Questionable Enable Register (QENR) which allows true conditions in the Questionable Event Register to be reported in the summary bit.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:QUESTIONABLE:ENABLE <bit_value> STATUS:QUESTIONABLE:ENABLE?
```

## Arguments

<bit\_value>::<NR1> is the enable mask of QENR. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the QENR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

STATUS:QUESTIONABLE:ENABLE256 enables the Calibration Summary bit.

STATUS:QUESTIONABLE:ENABLE? might return 256, showing that the bits in the QENR have the binary value 0000000100000000, which means that the Calibration Summary bit is valid.

## STATUS:QUESTIONABLE[:EVENT]? (Query Only)

Returns the contents of the Questionable Event Register (QEVr). Reading the QEVr clears it.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:QUESTIONABLE[:EVENT]?
```

## Arguments

None

## Returns

<NR1> is a decimal number showing the contents of the QEVr.

## Examples

STATUS:QUESTIONABLE:EVENT? might return 256, showing that the bits in the QEVr have the binary value 0000000100000000, which means that the Calibration Summary bit is set.

## STATus:QUESTionable:FREQuency:CONDition? (Query Only)

Returns the contents of the questionable frequency condition register.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

```
STATus:QUESTionable:FREQuency:CONDition?
```

### Arguments

None

### Returns

<NR1> is a decimal number showing the contents of the questionable frequency condition register.

### Examples

STATus:QUESTionable:FREQuency:CONDition? might return 512, showing that the bits in the questionable frequency condition register have the binary value 00000010 00000000, which means the Locked To External Ref bit is set.

## STATus:QUESTionable:FREQuency:ENABLE

Sets or queries the enable mask of the questionable frequency enable register which allows true conditions in the questionable frequency event register to be reported in the summary bit.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

```
STATus:QUESTionable:FREQuency:ENABLE <bit_value>  
STATus:QUESTionable:FREQuency:ENABLE?
```

### Arguments

<bit\_value>::<NR1> is the enable mask of the questionable frequency enable register. Range: 0 to 65535.

### Returns

<NR1> is a decimal number showing the contents of the questionable frequency enable register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

`STATUS:QUESTIONABLE:FREQUENCY:ENABLE512` enables the Locked To External Ref bit.

`STATUS:QUESTIONABLE:FREQUENCY:ENABLE?` might return 512, showing that the bits in the questionable calibration enable register have the binary value 00000010 00000000, which means that the Locked To External Ref bit is valid.

## STATus:QUESTionable:FREQuency[:EVENT]? (Query Only)

Returns the contents of the questionable frequency event register. Reading the register clears it.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

`STATus:QUESTionable:FREQuency[:EVENT]?`

### Arguments

None

### Returns

`<NR1>` is a decimal number showing the contents of the questionable frequency event register.

## Examples

`STATUS:QUESTIONABLE:FREQUENCY:EVENT?` might return 512, showing that the bits in the questionable frequency event register have the binary value 00000010 00000000, which means that the Locked To External Ref bit is set.

## STATus:QUESTionable:FREQuency:NTRansition

Sets or queries the negative transition filter value of the questionable frequency transition register.

### Conditions

Measurement views: All

### Group

Status commands

### Syntax

`STATus:QUESTionable:FREQuency:NTRansition <bit_value>`

`STATus:QUESTionable:FREQuency:NTRansition?`

### Arguments

`<bit_value>::<NR1>` is the negative transition filter value. Range: 0 to 65535.



## Returns

<NR1> is a decimal number showing the contents of the questionable frequency transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

STATUS:QUESTIONABLE:FREQUENCY:NTRANSITION#H0200 sets the negative transition filter value to #H0200.

STATUS:QUESTIONABLE:FREQUENCY:NTRANSITION? might return 512.

## STATus:QUESTionable:FREQuency:PTRansition

Sets or queries the positive transition filter value of the questionable frequency transition register.

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

STATus:QUESTionable:FREQuency:PTRansition <bit\_value>

STATus:QUESTionable:FREQuency:PTRansition?

## Arguments

<bit\_value>::<NR1> is the positive transition filter value. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the questionable frequency transition register. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION0 sets the positive transition filter value to 0.

STATUS:QUESTIONABLE:FREQUENCY:PTRANSITION? might return 0.

## STATus:QUESTionable:NTRansition

Sets or queries the negative transition filter value of the Questionable Transition Register (QTR).

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:QUESTIONABLE:NTRANSITION <bit_value> STATUS:QUESTIONABLE:NTRANSITION?
```

## Arguments

<bit\_value>::<NR1> is the negative transition filter value. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the QTR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

```
STATUS:QUESTIONABLE:NTRANSITION#H0020 sets the negative transition filter value to #H0020.
```

```
STATUS:QUESTIONABLE:NTRANSITION? might return 32.
```

## STATUS:QUESTIONABLE:PTRANSITION

Sets or queries the positive transition filter value of the Questionable Transition Register (QTR).

## Conditions

Measurement views: All

## Group

Status commands

## Syntax

```
STATUS:QUESTIONABLE:PTRANSITION <bit_value> STATUS:QUESTIONABLE:PTRANSITION?
```

## Arguments

<bit\_value>::<NR1> is the positive transition filter value. Range: 0 to 65535.

## Returns

<NR1> is a decimal number showing the contents of the QTR. Range: 0 to 32767 (The most-significant bit cannot be set true.)

## Examples

```
STATUS:QUESTIONABLE:PTRANSITION0 sets the positive transition filter value to 0.
```

```
STATUS:QUESTIONABLE:PTRANSITION? might return 0.
```

## STATUS:SEM:EVENTS? (Query Only)

Returns the current events and status conditions for the Spectral Emissions Mask measurement.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Status commands

## Syntax

```
STATus:SEM:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:SEM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:SGRAM[:MEASview<y>]:EVENTs (Query Only)

Returns the current events and status conditions for the spectrogram measurement.

## Conditions

Measurement views: Spectrogram

## Group

Status commands

## Syntax

```
STATus:SGRAM[:MEASview<y>]:EVENTs?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

## Returns

```
<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

`<info>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:SGRAM:MEASview1:EVENTs?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:SPECTrum[:MEASview<y>]:EVENTs? (Query Only)

Returns the current events and status conditions and status conditions for the spectrum measurement.

### Conditions

Measurement views: Spectrum

### Group

Status commands

### Syntax

`STATUS:SPECTrum[:MEASview<y>]:EVENTs?`

### Arguments

The parameter `<y>`=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default `y` is 1

### Returns

`<code>, "<edesc>[<info>]" {, <code>, "<edesc>[:<info>]" }`

Where

`<code>::<NR1>` is the error/event code (-32768 to 32767).

`<edesc>::<string>` is the description on the error/event.

`<info>::<string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:SPECTRUM:MEASVIEW2:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:SPURious:EVENTs ? (Query Only)

Returns the current events and status conditions for the Spurious measurement.

### Conditions

Measurement view: Spurious Display

### Group

Status commands

## Syntax

```
STATus:SPURious:EVENTs ?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

where

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <String>is the description of the error/event.

<einfo>:: <String>is the additional information on the error/event If there is no error, then response is 0, "No events to report."

## Examples

STATus:SPURIOUS:EVENTS ? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:SPURious:EVENTs ? (Query Only)

Returns the current events and status conditions for the Spurious measurement.

## Conditions

Measurement view: Spurious Display

## Group

Status commands

## Syntax

```
STATus:SPURious:EVENTs ?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

where

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <String>is the description of the error/event.

<einfo>:: <String>is the additional information on the error/event If there is no error, then response is 0, "No events to report."

## Examples

`STATUS:SPURIOUS:EVENTS ?` might return `12026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually.

## STATus:SQUality[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the signal quality measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Signal quality

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Signal quality).

## Group

Status commands

## Syntax

```
STATus:SQUality[:MEASview<y>]:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATus:SQUality:MEASview1:EVENTs?` might return `12026,"Acq Sampling Params: manual control"`, indicating that the sampling parameters are controlled manually for measurement view1.

## STATus:TDiagram[:MEASview<y>]:EVENTs? (Query only)

Returns the current events and status conditions for the trellis diagram measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Trellis diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Trellis diagram).

## Group

Status commands

## Syntax

```
STATus:TDIagram[:MEASview<y>]:EVENTs?
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error or event code (-32768 to 32767).

<edesc>::<string> is the description on the error or event.

<einfo>::<string> is the additional information on the error or event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:TDIagram:MEASview2:EVENTs? might return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:TXGain:EVENTs? (Query Only)

Queries events for the Transmission Gain display.

## Conditions

Measurement view: Transmission Gain

## Group

Status commands

## Syntax

```
STATus:TXGain:EVENTs?
```

## Returns

## Examples

```
STATUS:TXGAIN:EVENTS?
```

## STATUS:WLAN:AD:CONStE:EVENTs? (Query Only)

Queries the current events and status conditions for the Constellation measurement display.

### Conditions

Measurement view: 802.11ad Constellation

### Group

Status commands

### Syntax

STATUS:WLAN:AD:CONStE:EVENTs?

### Arguments

None

### Returns

<ecode>,"<edesc>[<einfo>]"{,<ecode>,"<edesc>[:<einfo>]"}

where

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <String>is the description of the error/event.

<einfo>:: <String>is the additional information on the error/event If there is no error, then response is 0, "No events to report."

### Examples

STATUS:WLAN:AD:CONStE:EVENTs? might return 12056,"Warm-up period", indicating the error code is 12056 and the description of that error code is "Warm up period".

## STATUS:WLAN:AD:EVM:EVENTs? (Query Only)

Returns the current events and status conditions for the EVM vs Time measurement.

### Conditions

Measurement views: 802.11ad EVM vs Time

### Group

Status commands

### Syntax

STATUS:WLAN:AD:EVM:EVENTs?

### Arguments

None



## Returns

```
<ecode>,"<edesc>[<einfo>]" {,<ecode>,"<edesc>[:<einfo>]" }
```

where

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <String> is the description of the error/event.

<einfo>:: <String> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATUS:WLAN:AD:EVM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:WLAN:AD:SUMMARY:EVENTs (Query Only)

Returns the current events and status conditions for the 802.11ad Summary display.

## Conditions

Measurement view: 802.11ad Summary

## Group

Status commands

## Syntax

```
STATUS:WLAN:AD:SUMMARY:EVENTs
```

## Arguments

None

## Returns

```
<ecode>,"<edesc>[<einfo>]" {,<ecode>,"<edesc>[:<einfo>]" }
```

where

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <String> is the description of the error/event.

<einfo>:: <String> is the additional information on the error/event. If there is no error, then response is 0, "No events to report."

## Examples

STATUS:WLAN:AD:SUMMARY:EVENTs might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:WLAN:AY:CONStE:EVENTs? (Query Only)

Query the current Events and Status conditions in the Constellation display.

## Conditions

Measurement view: 802.11ay Analysis Constellation

## Group

Status commands

## Syntax

STATus:WLAN:AY:CONStE:EVENTs?

## Returns

<ecode>, "<edesc> [<einfo>]", {<ecode>, "<edesc>[:<einfo>]"}

where:

<ecode> ::= <NR1> is the error/event code (Range: -32768 to 32767)

<edesc> ::= <string> is the description of the error/event.

<einfo> ::= <string> is the additional information on the error/event. If there are no errors, then the response will be 0, "No events to report."

## Examples

STAT:WLAN:AY:CONS:EVEN? may return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:AY:EVM:EVENTs? (Query Only)

Query the current events and status conditions for the EVM versus time measurement.

## Conditions

Measurement views: 802.11ay Analysis EVM

## Group

Status commands

## Syntax

STATus:WLAN:AY:EVM:EVENTs?

## Returns

<ecode>, "<edesc> [<einfo>]" {, <ecode>, "<edesc>[:<einfo>]"}

where:

<ecode> ::= <NR1> is the error/event code (-32768 to 32767).

<edesc> ::= <string> is the description of the error/event.

<einfo> ::= <string> is the additional information on the error/event. If there are no errors, the response is 0, "No events to report".

## Examples

`STAT:WLAN:AY:EVM:EVENT?` may return 12026, "Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:AY:SUMMary:EVENTs? (Query Only)

Query the current Events and Status conditions for the Summary measurement view.

### Conditions

Measurement view: 802.11ay Analysis Summary

### Group

Status commands

### Syntax

`STATus:WLAN:AY:SUMMary:EVENTs?`

### Returns

`<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}`

where:

`<ecode>::=<NR1>` is the Error/Event Code. Range: -32768 to 32767.

`<edesc>::=<string>` is the Description of the Error/Event.

`<einfo>::=<string>` is the additional information about the Error/Event. If there are no errors, then the response is 0, "No events to report".

### Examples

`STAT:WLAN:AY:SUMM:EVENT?` may return 12026, "Acq Sampling Params: manual control", indicating that sampling parameters are controlled manually.

## STATus:WLAN:CONSte:EVENTs? (Query Only)

Returns the current instrument status concatenated with WLAN constellation-specific event information.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Status commands

### Syntax

`STATus:WLAN:CONSte:EVENTs?`

## Returns

ASCII string representation of instrument status as follows:

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:WLAN:CONSTE:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:WLAN:CRESponse:EVENTs? (Query Only)

Returns the current events and status conditions for the WLAN Channel Response display.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Status commands

## Syntax

```
STATUS:WLAN:CRESponse:EVENTs?
```

## Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:WLAN:CRES:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATUS:WLAN:EVM:EVENTs? (Query Only)

Returns the current instrument status concatenated with WLAN EVM-specific event information.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Status commands

## Syntax

```
STATus:WLAN:EVM:EVENTs?
```

## Returns

ASCII string representation of instrument status as follows:

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATus:WLAN:EVM:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:MERRor:EVENTs? (Query Only)

Returns the current instrument status concatenated with WLAN Magnitude Error-specific event information.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Status commands

## Syntax

```
STATus:WLAN:MERRor:EVENTs?
```

## Returns

```
<ecode>,"<edesc>[<einfo>]"{"<ecode>,"<edesc>[:<einfo>]"}
```

Where:

<ecode>:: <NR1> is the error/event code (-32768 to 32767).

<edesc>:: <string> is the description on the error/event.

`<einfo>:: <string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:WLAN:MERR:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:PERRor:EVENTs? (Query Only)

Returns instrument status concatenated with WLAN Phase Error-specific event information.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Status commands

### Syntax

`STATus:WLAN:PERRor:EVENTs?`

### Returns

ASCII string representation of instrument status as follows:

```
<ecode>, "<edesc> [<einfo>] " { , <ecode>, "<edesc> [ :<einfo>] " }
```

Where

`<ecode>:: <NR1>` is the error/event code (-32768 to 32767).

`<edesc>:: <string>` is the description on the error/event.

`<einfo>:: <string>` is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

`STATUS:WLAN:PERR:EVENTS?` might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:PVTime:EVENTs? (Query Only)

Returns the current instrument status concatenated with WLAN Power vs. Time-specific event information.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Status commands

## Syntax

ASCII string representation of instrument status as follows:

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event.

If there is no error, the response is 0, "No events to report".

## Examples

STATUS:WLAN:PVT:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:STABLE:EVENTs? (Query Only)

Returns instrument status concatenated with WLAN Symbol Table-specific event information.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Status commands

## Syntax

```
STATus:WLAN:STABLE:EVENTs?
```

## Returns

ASCII string representation of instrument status as follows:

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[:<einfo>]" }
```

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event. If there is no error, the response is 0, "No events to report".

## Examples

STATUS:WLAN:STABLE:EVENTS? might return 12026,"Acq Sampling Params: manual control", indicating that the sampling parameters are controlled manually.

## STATus:WLAN:SUMMARY:EVENTs? (Query Only)

Returns the current events and status conditions for the WLAN summary measurement.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Status commands

### Syntax

```
STATus:WLAN:SUMMARY:EVENTs?
```

### Arguments

None

### Returns

```
<ecode>, "<edesc>[<einfo>]" {, <ecode>, "<edesc>[<einfo>]"}
```

Where

<ecode>::<NR1> the error/event code (-32768 to 32767).

<edesc>::<string> the description on the error/event.

<einfo>::<string> the additional information on the error/event. If there is no error, the response is 0, "No events to report".

Returns the contents of the Status Byte Register (SBR) in the status/event reporting structure using the Master Summary Status (MSS) bit. Refer to Section 3, *Status and Events*, for the register information.

## \*STB? (Query Only)

Returns the contents of the Status Byte Register (SBR) in the status/event reporting structure using the Master Summary Status (MSS) bit. Refer to Section 3, *Status and Events*, for the register information.

### Conditions

Measurement views: All

### Group

IEEE common commands

### Syntax

```
*STB?
```

### Related Commands

```
*CLS, *ESE, *ESR?, *SRE
```



## Arguments

None

## Returns

<NR1> representing the contents of the SBR as a decimal number.

## Examples

\*STB? might return 96, indicating that the SBR contains binary 0110 0000.

## System commands

### SYSTEM:ANTenna:AMPLifier? (Query Only)

Queries the status of the antenna amplifier.

#### Conditions

Requires an RSA306B/500A/600A instrument with an antenna.

#### Group

System commands

#### Syntax

```
SYSTEM:ANTenna:AMPLifier?
```

#### Returns

1 or ON means amplifier is on. 0 or OFF means amplifier is off.

### SYSTEM:ANTenna:CAPabilities:REPort? (Query Only)

Queries the selected antenna capabilities for the items.

#### Conditions

Requires an RSA306B/500A/600A instrument with an antenna.

#### Group

System commands

#### Syntax

```
SYSTEM:ANTenna:CAPabilities:REPort?
```

#### Returns

ReportsNothing = 1, ReportsAzimuth = 2, ReportsElevation = 4,

ReportsActiveMode = 8, ReportsBandChange = 16, ReportsRollAngle = 32,  
ReportsButtonPresses = 64

## Examples

SYSTEM:ANTENNA:CAPABILITIES:REPORT? might return 126 when the set to Simulator. This is 64+32+16+8+4+2.

SYSTEM:ANTENNA:CAPABILITIES:REPORT? might return 6, which indicates capabilities of azimuth and elevation reporting.

## SYSTEM:ANTenna:CATalog? (Query Only)

Queries the catalog of antennas. Use the SYSTem:ANTenna:SElected command to select an antenna from the catalog.

### Conditions

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTem:ANTenna:CATalog?
```

### Returns

<string>, <string> is a list of antennas in the catalog.

## Examples

SYSTEM:ANTENNA:CATALOG? might return "Alaris DF-A0047","Simulator".

## SYSTEM:ANTenna:CONNect (No Query Form)

Initiates antenna connection.

### Conditions

Requires antenna or antenna simulator mode. Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTem:ANTenna:CONNect { 1 | 0 }
```

### Arguments

1 or ON connects antenna.

0 or OFF does not connect antenna.

## SYSTEM:ANTenna:DATA? (Query Only)

Queries the true north azimuth, uncorrected compass bearing, elevation, and roll values of the antenna.

### Conditions

Antenna mode must be set to Simulator to report roll value.

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTEM:ANTenna:DATA?
```

### Related Commands

```
SYSTEM:ANTenna:SETDATA
```

### Returns

```
<numeric>::<NR3>
```

```
<numeric>,<numeric>,<numeric>,<numeric>
```

### Examples

SYSTEM:ANTENNA:DATA? might return 65.6509800000,54.2109800000,37.8086400000 indicating the true north azimuth is 65.6°, the uncorrected compass bearing is 54.2°, and the elevation is 37.8°.

## SYSTEM:ANTenna:DECLination? (Query Only)

Queries current calculated declination.

### Conditions

Requires an RSA306B/500A/600A instrument with an antenna.

### Group

System commands

### Syntax

```
SYSTEM:ANTenna:DECLination?
```

### Returns

<NR2> is the calculated declination.

### Examples

SYSTEM:ANTENNA:DECLINATION? might return 10.0, indicating that the calculated declination is 10.0°.

## **SYSTem:ANTenna:DECLination:USER**

Sets or queries the manually entered declination.

### **Conditions**

Requires an RSA306B/500A/600A instrument with an antenna.

### **Group**

System commands

### **Syntax**

```
SYSTem:ANTenna:DECLination:USER <NR2> SYSTem:ANTenna:DECLination:USER?
```

### **Related Commands**

[SYSTem:ANTenna:DECLination:USER:STATe](#)

### **Arguments**

<NR2> is the manually entered declination.

### **Returns**

<NR2> is the manually entered declination.

### **Examples**

```
SYSTEM:ANTENNA:DECLINATION:USER?
```

might return 10.0000000000, indicating that the manually forced declination is set to 10.0°.

## **SYSTem:ANTenna:DECLination:USER:STATe**

Sets or queries to force declination.

### **Conditions**

Requires an RSA306B/500A/600A instrument with an antenna.

### **Group**

System commands

### **Syntax**

```
SYSTem:ANTenna:DECLination:USER:STATe { 1 | 0 | ON | OFF }  
SYSTem:ANTenna:DECLination:USER:STATe?
```

### **Related Commands**

[SYSTem:ANTenna:DECLination:USER](#)

## Arguments

1 or ON forces the declination to a user entered value.

0 or OFF sets the declination to not be forced.

## Returns

1 or ON means the declination is set to a manually entered value.

0 or OFF means the declination is not forced.

## SYSTEM:ANTenna:FREQuency:BAND? (Query Only)

Queries the currently selected frequency band for the antenna, assuming that it has the ability to report back which band is selected.

## Conditions

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTEM:ANTenna:FREQuency:BAND?
```

## Returns

It returns 0 for all currently available antennas (and the Simulator), because they do not have the ability to report back which band is selected.

## Examples

```
SYSTEM:ANTENNA:FREQUENCY:BAND?
```

## SYSTEM:ANTenna:MANufacturer? (Query Only)

Queries the antenna manufacturer.

## Conditions

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTEM:ANTenna:MANufacturer?
```

## Returns

<string>

## Examples

SYSTEM:ANTENNA:MANUFACTURER? might return "Alaris".

## SYSTEM:ANTenna:MODEL? (Query Only)

Queries the antenna model.

### Conditions

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

SYSTEM:ANTenna:MODEL?

### Returns

<string>

## Examples

SYSTEM:ANTENNA:MODEL? might return "DF-A0047".

## SYSTEM:ANTenna:OPTions? (Query Only)

Queries the antenna options.

### Conditions

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

SYSTEM:ANTenna:OPTions?

### Returns

<string>

## Examples

SYSTEM:ANTENNA:OPTIONS? might return "None".

---

## SYSTem:ANTenna:SElected

Sets or queries the selected antenna.

### Conditions

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTem:ANTenna:SElected <string> SYSTem:ANTenna:SElected?
```

### Related Commands

[SYSTem:ANTenna:CATalog?](#)

### Arguments

<string> is an item from the antenna catalog.

Alaris DF-A0047 sets analyzer to Alaris antenna. Simulator sets the analyzer to the simulated antenna.

### Returns

“None” means the analyzer is set to no antenna.

“Alaris DF-A0047” means the analyzer is set to the Alaris antenna. “Simulator” means the analyzer is set to the simulated antenna.

### Examples

SYSTem:ANTenna:SElected? might return “Alaris DF-A0047”, indicating that the antenna selection is set to the Alaris DF-A0047.

## SYSTem:ANTenna:SERial? (Query Only)

Queries the antenna serial number.

### Conditions

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTem:ANTenna:SERial?
```

### Returns

<string>

## Examples

SYSTEM:ANTENNA:SERIAL? might return "A22500".

## SYSTEM:ANTenna:SETDATA (No Query Form)

Sets the amplifier, true north azimuth, uncorrected compass bearing, elevation, and roll values of the antenna.

### Conditions

Antenna mode must be set to Simulator to report roll value.

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTEM:ANTenna:SETDATA
```

```
<boolean>,<numeric>,<numeric>,<numeric>,<numeric>
```

### Related Commands

```
SYSTEM:ANTenna:DATA?
```

### Arguments

<boolean> is { 1 | 0 | ON | OFF }

<numeric>::<NR3>

### Examples

```
SYSTEM:ANTENNA:SETDATA 1,65.6509800000,54.2109800000,37.8086400000
```

will set the amplifier to ON (1), the true north azimuth to 65.6°, the uncorrected compass bearing to 54.2°, and the elevation to 37.8°.

## SYSTEM:BATTery:CCYCLE? (Query Only)

Queries the battery charge cycle.

### Conditions

Requires an RSA500A series instrument with a battery installed.

### Group

System commands

### Syntax

```
SYSTEM:BATTery:CCYCLE?
```



## Returns

Returns the number of charge cycles of the battery.

## Examples

`SYSTEM:BATTERY:CCYCLE?` might return 10, indicating that the battery has been charged 10 times.

## SYSTEM:BATTERY:DATE? (Query Only)

Queries the battery manufactured date.

## Conditions

Requires an RSA500A series instrument with a battery installed.

## Group

System commands

## Syntax

`SYSTEM:BATTERY:DATE?`

## Returns

Returns the date string as yyyy-mm-dd.

## Examples

`SYSTEM:BATTERY:DATE?` might return "2015-10-05", indicating the battery manufactured date is October 5, 2015.

## SYSTEM:BATTERY:RCHARGE? (Query Only)

Queries the relative charge remaining.

## Conditions

Requires an RSA500A series or RSA600A series instrument with a battery installed.

## Group

System commands

## Syntax

`SYSTEM:BATTERY:RCHARGE?`

## Returns

Returns the relative charge remaining in percent.

## Examples

`SYSTEM:BATTERY:RCHARGE?` might return 50, indicating the relative charge remaining is 50%.

## **SYSTem:BATTeRy:SERial? (Query Only)**

Queries the battery serial number.

### **Conditions**

Requires an RSA500A series instrument with a battery installed.

### **Group**

System commands

### **Syntax**

SYSTem:BATTeRy:SERial?

### **Returns**

Returns the battery serial number.

### **Examples**

SYSTem:BATTeRy:SERial? might return 1234, indicating the serial number is 1234.

## **SYSTem:BATTeRy:STATus? (Query Only)**

Queries the battery status.

### **Conditions**

Requires an RSA500A series instrument with a battery installed.

### **Group**

System commands

### **Syntax**

SYSTem:BATTeRy:STATus??

### **Returns**

ACFull means AC present and fully charged ACCharging means AC present and charging ACNBattery means AC present and no battery  
FULL means battery fully charged

EMPTy means battery fully discharged DISCharging battery discharging ERRor battery error

### **Examples**

SYSTem:BATTeRy:STATus?? might return ACNB, indicating no battery is installed.

## **SYSTem:BATTeRy:TEMPerature? (Query Only)**

Queries the battery temperature.

## Conditions

Requires an RSA500A series instrument with a battery installed.

## Group

System commands

## Syntax

```
SYSTem:BATTery:TEMPerature?
```

## Returns

Returns the battery temperature in degrees Celsius.

## Examples

SYSTEM:BATTERY:TEMPERATURE? might return 15, indicating that the battery temperature is 15 °C.

## SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Sets or queries the GPIB address of the instrument.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <value>  
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?
```

## Arguments

<value> : : <NR1> specifies the GPIB address of the instrument.

\*RST has no effect on the value.

## Examples

SYSTEM:COMMUNICATE:GPIB:SELF:ADDRESS 18 sets the GPIB address to 18.

## SYSTem:DATE

Sets or queries the date (year, month, and day). This command is equivalent to the date setting through the Windows Control Panel.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:DATE <year>,<month>,<day> SYSTem:DATE?

## Related Commands

[SYSTem:TIME](#)

## Arguments

<year>::<NRf> specifies the year (4 digits). Range: 2000 to 2099.

<month>::<NRf> specifies the month. Range: 1 (January) to 12 (December).

<day>::<NRf> specifies the day. Range: 1 to 31. These values are rounded to the nearest integer.

\*RST has no effect on the settings.

## Examples

SYSTEM:DATE2008,3,19 sets the internal calendar to March 19, 2008.

## SYSTem:ERRor:ALL? (Query Only)

Queries the error/event queue for all the unread items and removes them from the queue. The response is a comma separated list of number, string pairs in FIFO order. For details of the error messages, refer to (See Table 3-16.)

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:ERRor:ALL?

## Arguments

None

## Returns

<ecode>,"<edesc>[;<einfo>]"{"<ecode>,"<edesc>[;<einfo>]"}

Where

<ecode>::<NR1> is the error/event code (-32768 to 32767).

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the detail of the error/event.

If the queue is empty, the response is 0, "No error; Queue empty - No events to report".

## Examples

`SYSTEM:ERROR:ALL?` might return -130, "Suffix error; Unrecognized suffix, INPutMLEVel -10dB", , indicating that the unit of the reference level is improper.

## SYSTEM:ERROR:CODE:ALL? (Query Only)

Queries the error/event queue for all the unread items and removes them from the queue. The response returns a comma separated list of only the error/event code numbers in FIFO order. For details of the error messages, refer to (See Table 3-16.)

## Conditions

Measurement views: All

## Group

System commands

## Syntax

`SYSTEM:ERROR:CODE:ALL?`

## Arguments

None

## Returns

`<ecode>{ , <ecode>}`

Where

`<ecode> : : <NR1>` is the error/event code, ranging from -32768 to 32767.

If the queue is empty, the response is 0.

## Examples

`SYSTEM:ERROR:CODE:ALL?` might return -101,-108 of the error codes.

## SYSTEM:ERROR:CODE[:NEXT]? (Query Only)

Queries the error/event queue for the next item and removes it from the queue. The response returns only the error/event code number omitting the string. Except for the shortened response, the query operates identically to `SYSTEM:ERROR[:NEXT]?`. For details of the error messages, refer to (See Table 3-16.)

## Conditions

Measurement views: All

## Group

System commands

## Syntax

`SYSTem:ERRor:CODE[:NEXT]?`

## Arguments

None

## Returns

`<ecode> : : <NR1>` is the error/event code, ranging from -32768 to 32767.

## Examples

`SYSTEM:ERROR:CODE:NEXT?` might return -101 of the error code.

## SYSTem:ERRor:COUNT? (Query Only)

Queries the error/event queue for the number of unread items. As errors and events may occur at any time, more items may be present in the queue at the time it is actually read.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

`SYSTem:ERRor:COUNT?`

## Arguments

None

## Returns

`<enum> : : <NR1>` is the number of errors/events.

If the queue is empty, the response is 0.

## Examples

`SYSTEM:ERROR:COUNT?` might return 2, indicating that the error/event queue contains two of unread errors/events.

## SYSTem:ERRor[:NEXT]? (Query Only)

Queries the error/event queue for the next item and removes it from the queue. The response returns the full queue item consisting of an integer and a string. For details of the error messages, refer to the Measurement status conditions table in the PDF of the Programmer manual for your product.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:ERRor[:NEXT]?
```

## Arguments

None

## Returns

```
<ecode>, "<edesc>[;<einfo>]"
```

Where

<ecode>::<NR1> is the error/event code, ranging from -32768 to 32767.

<edesc>::<string> is the description on the error/event.

<einfo>::<string> is the additional information on the error/event.

## Examples

SYSTEM:ERROR:NEXT? might return -130, "Suffix error; Unrecognized suffix, INPutMLEVel -10dB", indicating that the unit is improper.

## SYSTem:GNSS:ALTitude (No Query Form)

Query the GNSS altitude.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTem:GNSS:ALTitude <altitude>
```

## Returns

<altitude>::<NRF> in meters.

## Examples

SYSTEM:GNSS:ALTITUDE might return 60.1, the GNSS altitude.

## **SYSTEM:GNSS:COURSE? (Query Only)**

Query the GNSS course in degrees.

### **Conditions**

Measurement views: All

Requires an RSA306B/500A/600A instrument.

### **Group**

System commands

### **Syntax**

SYSTEM:GNSS:COURSE?

### **Examples**

SYSTEM:GNSS:COURSE? might return 162.78 indicating the GNSS course is 162.78 degrees.

## **SYSTEM:GNSS:DATA? (Query Only)**

Query the GNSS data.

### **Conditions**

Measurement views: All

Requires an RSA306B/500A/600A instrument.

### **Group**

System commands

### **Syntax**

SYSTEM:GNSS:DATA?

### **Returns**

<status>, <longitude>, <latitude>, <altitude>, <timestamp>, <speed>, <course>, satellites <HDOP>

### **Examples**

SYSTEM:GNSS:DATA? might return

LOCK,-122.8197333333,45.4994000000,63.4000000000,"2012-06-01T13:51:26.00",22.96480

## **SYSTEM:GNSS:HDOP? (Query Only)**

Query the GNSS HDOP value.



---

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTem:GNSS:HDOP?
```

## Returns

<value> : : <NR1> the horizontal DOP value.

## Examples

SYSTEM:GNSS:HDOP? might return 900.000E-3, the GNSS HDOP value.

## SYSTEM:GNSS:LATitude? (Query Only)

Query the GNSS latitude.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTem:GNSS:LATitude?
```

## Examples

SYSTEM:GNSS:LATITUDE? might return 45.497193, the GNSS latitude.

## SYSTEM:GNSS:LONGitude? (Query Only)

Query the GNSS longitude.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

`SYSTEM:GNSS:LONGitude?`

## Examples

`SYSTEM:GNSS:LONGITUDE?` might return the -122.82372, the GNSS longitude.

## SYSTEM:GNSS:SATellites? (Query Only)

Query the number of GNSS satellites.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

`SYSTEM:GNSS:SATellites?`

## Examples

`SYSTEM:GNSS:SATELLITES?` might return 10.000 indicating the number of GNSS satellites is 10.

## SYSTEM:GNSS:SElected

Selects or queries the source for the GNSS signal.

The `INTernal` argument is not available with the RSA306.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

`SYSTEM:GNSS:SElected { NONE | INTernal | EXTernal }`

`SYSTEM:GNSS:SElected?`

## Arguments

`NONE`: selects no GNSS system.

`EXTernal`: This can be an external USB device or other external device on the table.

## Returns

NONE: No GNSS system is selected.

EXTERNAL: the external GNSS system.

## Examples

SYSTEM:GNSS:SELECTED? might return EXT indicating the source for the GNSS system is an external device.

## SYSTEM:GNSS:SERIAL:BAUD

Sets or queries the GNSS serial baud rate.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTEM:GNSS:SERIAL:BAUD { 4800, 9600, 19200, 38400 }
```

```
SYSTEM:GNSS:SERIAL:BAUD?
```

## Examples

SYSTEM:GNSS:SERIAL:BAUD 19200 sets the baud rate to 19200.

## SYSTEM:GNSS:SERIAL:PORT

Sets or queries the GNSS serial communications port.

## Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

## Group

System commands

## Syntax

```
SYSTEM:GNSS:SERIAL:PORT <COM_Port> SYSTEM:GNSS:SERIAL:PORT?
```

## Examples

SYSTEM:GNSS:SERIAL:PORT 1 sets PORT 1 as the GNSS serial communications port.

## **SYSTEM:GNSS:SPEEd? (Query Only)**

Query the GNSS speed in meter/hour.

### **Conditions**

Measurement views: All

Requires an RSA306B/500A/600A instrument.

### **Group**

System commands

### **Syntax**

SYSTEM:GNSS:SPEEd?

### **Examples**

SYSTEM:GNSS:SPEED? might return 31.33583, indicating the GNSS speed is 31.33583 meters/hour.

## **SYSTEM:GNSS:STATus? (Query Only)**

Query the GNSS status.

### **Conditions**

Measurement views: All

Requires an RSA306B/500A/600A instrument.

### **Group**

System commands

### **Syntax**

SYSTEM:GNSS:STATus?

### **Returns { OFF**

| BAD | UNLock | LOCKed }

Where:

OFF - the GNSS is turned off.

BAD - a communication error.

UNL - the GNSS is unlocked.

LOCK - the GNSS is locked.

### **Examples**

SYSTEM:GNSS:STATUS? might return BAD indicating a communication error has occurred. For example, the GNSS port or baud rate is wrong.

## SYSTem:GNSS:TIMestamp? (Query Only)

Query the GNSS timestamp.

### Conditions

Measurement views: All

Requires an RSA306B/500A/600A instrument.

### Group

System commands

### Syntax

```
SYSTem:GNSS:TIMestamp?
```

### Examples

SYSTEM:GNSS:TIMESTAMP? might return 2012-06-01T13:47:21.00, the GNSS timestamp.

## SYSTem:KLOCK

Enables or disables the local lockout operation.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTem:KLOCK { OFF | ON | 0 | 1 } SYSTem:KLOCK?
```

### Arguments

OFF or 0 disables the local lockout operation.

ON or 1 enables the local lockout operation.

### Examples

SYSTEM:KLOCK ON enables the local lockout operation.

## SYSTem:LINK:SETTings

Sets or queries whether or not global settings linking enabled for the measurements displayed.

### Conditions

Measurement views: General Signal viewing and GPDemod Measurements.

## Group

System commands

## Syntax

```
SYSTem:LINK:SETTings
```

## Arguments

0 indicates global settings linking is disabled.

1 indicates global settings linking is enabled.

## Examples

`SYSTem:LINK:SETTings 1` enables the global settings linking for the displayed measurement views.

## SYSTem:LINK:SETTings:SYNC

Sets the synchronize settings referring to source for the measurements displayed.

## Conditions

Measurement views: General Signal viewing and GPDemod Measurements.

## Group

System commands

## Syntax

```
SYSTem:LINK:SETTings:SYNC { 'Source1' | 'Source2' | 'Source3' | 'Source4' }
```

## Arguments

'Source1' or 'Source2'.

## Examples

`SYSTem:LINK:SETTings:SYNC 'Source2'` synchronizes the settings referring to Source2 for the displayed measurement views.

## SYSTem:OPTions? (Query Only)

Queries the options installed in the analyzer. This command is equivalent to the IEEE common command `*OPT?`.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:OPTions?

## Arguments

None

## Returns

<option>::<string> contains the comma-separated option numbers.

## Examples

SYSTEM:OPTIONS? might return "01,02,20", indicating that Option 01, 02, and 20 are currently installed in the analyzer.

## SYSTem:PRESet (No Query Form)

Restores the analyzer to the defaults. This command is equivalent to the Preset key on the front panel.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:PRESet

## Arguments

None

## Examples

SYSTEM:PRESET restores the analyzer to the defaults.

## SYSTem:PRESet:APPLication:ACTion

Sets or queries the preset action for the application preset type.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:PRESet:APPLication:ACTion { RECall | SHOW }  
SYSTem:PRESet:APPLication:ACTion?
```

## Arguments

RECall recalls the selected preset for the application preset type.

SHOW lists the available presets for the application preset type.

## Examples

SYSTEM:PRESET:APPLICATION:ACTION SHOW lists the available presets for the application preset type.

## SYSTem:PRESet:BLUEtooth:STANdard

Sets or queries the standard, setup, Retaining Current Reference level, and Retaining Center Frequency in the Bluetooth standard preset.

## Conditions

Measurement views: Launch BT Application

## Group

System commands

## Syntax

```
SYSTem:PRESet:BLUEtooth:STANdard {BR | LE | EDR}, {BW20DB | MODD | INBE | NCOM}, {1  
| 0}, {1 | 0}
```

## Arguments

BR is the Basic Rate.

LR is the Low Energy Rate.

EDR is the Enhanced Data Rate.

BR is the Basic Rate.

BR is the Basic Rate.

<test\_setup> ::= {BW20DB|MODD|INBE|NCOM} the name of the test setup.

<Retaining\_Current\_Reflevel> ::= [1 | 0] where 1 indicates this value is checked in the view; 0 indicates the value is unchecked. .

<Retaining\_Center\_Frequency> ::= [1 | 0] where 1 indicates this value is checked in the view; 0 indicates the value is unchecked.

## Examples

SYSTEM:PRESET:BLUETOOTH:STANDARD BR,MODD,0,1 might return will set Preset to Bluetooth, standard to basic rate, Test Setup to Modulation/Frequency Offset/Drift/Output Power, disable the retain current center frequency setting, Enable the Retain current ref level setting.



---

## SYSTem:PRESet:DPX (No Query Form)

Restores the analyzer to the defaults for the DPX preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTem:PRESet:DPX { SWEpt | REALtime | ZERospan }
```

### Arguments

SWEpt restores the setup of the Swept DPX measurement.

REALtime restores the setup of the Realtime DPX measurement (110 MHz span).

ZERospan restores the setup of the Zero Span DPX measurement.

### Examples

```
SYSTEM:PRESET:DPX REALtime restores the DPX, Spectrum to realtime.
```

## SYSTem:PRESet:DPX:ACTion

Sets or queries the preset action for the DPX preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTem:PRESet:DPX:ACTion { RECall | SHOW } SYSTem:PRESet:DPX:ACTion?
```

### Arguments

RECall recalls the selected preset for the DPX preset type.

SHOW lists the available presets for the DPX preset type.

### Examples

```
SYSTEM:PRESET:DPX:ACTION SHOW lists the available presets for the DPX preset type.
```

## **SYSTem:PRESet:DPX:SElected**

Sets or queries the presets for the DPX preset type.

### **Conditions**

Measurement views: All

### **Group**

System commands

### **Syntax**

```
SYSTem:PRESet:DPX:SElected { OPEn | SWEpt | REALtime | ZERospan }  
SYSTem:PRESet:DPX:SElected?
```

### **Arguments**

`OPEn` sets the DPX measurement to Open the DPX display.

`SWEpt` sets the DPX measurement to Swept mode.

`REALtime` sets the DPX measurement to Real-time mode (110 MHz span).

`ZERospan` sets the DPX measurement to Zero Span mode.

### **Examples**

```
SYSTEM:PRESET:DPX:SELECTED REALtime sets the DPX, Spectrum to realtime
```

## **SYSTem:PRESet:LTE:STANdard**

Sets or queries the following: Preset, Test Setup, frame structure channel bandwidth, Base Station category (for LTE ACLR and SEM test setups), Adjacent

channel type (for LTE ACLR test setup), Retain current Center Frequency setting, and Retain current Reference Level.

### **Conditions**

Measurement view: Standards Presets

### **Group**

System commands

### **Syntax**

```
SYSTem:PRESet:LTE:STANdard { MODulation | ACLR | CS | SEM, FDD | TDD, BW1PT4M  
| BW3M | BW5M | BW10M | BW15M | BW20M, WACA | WACB1 | WACB2 | HACAB |  
LACAB | MACAB, UTRA | EUTRA, CR1PT28M | CR3PT84M | CR7PT68M, 0 | 1, 0 | 1 ]  
SYSTem:PRESet:LTE:STANdard?
```

## Arguments

MODulation | ACLR | CS | SEM, FDD | TDD, BW1PT4M | BW3M | BW5M | BW10M | BW15M | BW20M, WACA | WACB1 | WACB2 | HACAB | LACAB | MACAB, UTRA | EUTRA, CR1PT28M | CR3PT84M | CR7PT68M, 0 | 1, 0 | 1

## Returns

See Arguments.

## Examples

SYSTEM:PRESET:LTE:STANDARD? might return MOD,FDD,BW1PT4M,0,1, indicating that the test setup is Cell ID, Frame Structure, Channel BW, Retain Center Frequency, Retain Reference Level.

## SYSTEM:PRESet[:MAIN] (No Query Form)

Restores the analyzer to the defaults. This command is equivalent to the SYSTem:PRESet command.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:PRESet[:MAIN] { MAIN| } SYSTem:PRESet[:MAIN] ?

## Arguments

CURRent restores the analyzer to the current version of the main preset.

V1 restores the analyzer to the original version of the main preset.

V2 restores the analyzer to the Full Spectrum version of the main preset.

## Examples

SYSTEM:PRESET[:MAIN] CURRENT restores the analyzer to the defaults of the current version.

## SYSTEM:PRESet:MAIN:ACTion

Sets or queries the preset action for the main preset type.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:PRESet:MAIN:ACTion { RECall | SHOW } SYSTem:PRESet:MAIN:ACTion?
```

## Arguments

RECall recalls the selected preset for the main preset type.

SHOW lists the available presets for the main preset type.

## Examples

```
SYSTEM:PRESET:MAIN:ACTION SHOW lists the available presets for the main preset type.
```

## SYSTem:PRESet:MAIN:SElected

Sets or queries the presets for the Main preset type.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:PRESet:MAIN:SElected { CURRent | V1 | V2 } SYSTem:PRESet:MAIN:SElected?
```

## Arguments

CURRent selects the current version of the Main presets.

V1 selects the previous version of the main preset.

V2 selects Full Spectrum Sweep version of the current main preset.

## Examples

```
SYSTEM:PRESET:MAIN:SELECTED CURRent selects the current version of the Main presets.
```

## SYSTem:PRESet:NR:STANdard

It Sets or queries the 5G NR standard, retain the current center frequency setting and Retain current ref level setting in the 5G NR standard preset.

## Conditions

Measurement view: All 5G NR measurement view

## Group

System commands

## Syntax

```
SYSTem:PRESet:NR:STANdard
```

```
<TestSetup>,<Link Direction>,<No of CC>,<Ch Bandwidth>,<TestModel>< Retain
current center Frequency setting>,<Retain Current Ref level setting>.
```

```
SYSTem:PRESet:NR:STANdard?
```

In order to perform Preset operation, have to execute the following commands sequentially.

## Examples

```
SYSTem:PRESet:NR:STANdard CONS,UPLI,1,10000000,THIRTY,0,1
```

```
SYSTem:PRESet:STANdards:SELEcted FIVEGNR
```

```
SYSTem:PRESet:STANdards FIVEGNR
```

## Arguments

MOD | ACP | CHP | SEM, OBW, EVM DOWNlink | UPLink, 1 to 8, 5000000 to 100000000, FIFTEEN | THIRTY | SIXTY | ONE20, TM1DOT1 | TM1DOT2 | TM2 | TM2A | TM3DOT1 | TM3DOT1A | TM3DOT2 | TM3DOT3, 0 | 1, 0 | 1

This command sets the Preset to 5GNR, standard to NR Constellation, Uplink, single carrier, Channel Bandwidth to 100MHz, sub carrier spacing to 30 kHz, disables the retain current center frequency setting, enables the Retain current ref level setting.

### Note:



- Sub carrier spacing parameter need to applied only on Modulation and EVM vs Symbol Preset
- Link direction doesn't applicable on PVT measurement.

## SYSTem:PRESet:P25:STANdard (No Query Form)

This command restores the analyzer defaults for the P25 Standards preset.

## Conditions

Measurement view: Any P25 measurement

## Group

System commands

## Syntax

```
SYSTem:PRESet:P25:STANdard <{ PHASe1 | PHASe2 }>,<{ C4FM | HCPM | HDQPSK }>,<{ 1 | 0 }>
```

## Arguments

PHASe1 and PHASe2 are the P25 standard types.

C4FM, HCPM, HDQPSK are the P25 modulation types.

1 = retain center frequency; 0 = do not retain center frequency.

## Returns

<standard> is the P25 standard type: Phase1, Phase2.

<modulation type> is the P25 modulation type: C4FM, HCPM (Inbound), HDQPSK (Outbound).

<{ 1 | 0 }> controls whether or not the center frequency is retained: 0 = not retained; 1 = retained.

## Examples

SYSTEM:PRESET:P25:STANDARD? might return PHASE1,C4FM,0, indicating that the P25 preset is set to standard type Phase 1, modulation type C4FM, and that the center frequency is set to not be retained.

## SYSTEM:PRESet:STANdards (No Query Form)

Restores the analyzer to the defaults for the WLAN Standards preset type. To restore the analyzer to a preset file, use the command SYSTEM:PRESet:USER.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTEM:PRESet:STANdards <WLAN>
```

### Arguments

<WLAN> specifies to restore the analyzer to the WLAN standards preset type.

### Examples

SYSTEM:PRESET:STANDARDS "WLAN" restores the analyzer to the defaults for the WLAN Standards preset type.

## SYSTEM:PRESet:STANdards:ACTion

Sets or queries the preset action for the Standards preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTEM:PRESet:STANdards:ACTion { RECall | SHOW } SYSTEM:PRESet:STANdards:ACTion?
```

### Arguments

RECall recalls the selected preset for the Standards preset type.

SHOW lists the available presets for the Standards preset type.

## Examples

`SYSTEM:PRESET:STANDARDS:ACTION SHOW` lists the available presets for the Standards preset type.

## SYSTem:PRESet:USER (No Query Form)

Restores the analyzer to the defaults for the user preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

`SYSTem:PRESet:USER <file_name>`

### Arguments

`<file_name>::<string>` specifies the user presets file to restore. The file extension is `.Setup`. You can omit the extension.

## Examples

`SYSTEM:PRESET:USER "Spectrum"` restores the defaults of the file "Spectrum.Setup" from the C:\RSA5100B Files\User Presets directory.

## SYSTem:PRESet:USER:ACTIon

Sets or queries the preset action for the user preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

`SYSTem:PRESet:USER:ACTIon { RECall | SHOW } SYSTem:PRESet:USER:ACTIon?`

### Arguments

`RECall` recalls the selected preset for the user preset type.

`SHOW` lists the available presets for the user preset type.

## Examples

`SYSTEM:PRESET:USER:ACTION SHOW` lists the available presets for the User preset type.

## SYSTem:PRESet:USER:SElected

Sets or queries the presets for the User preset type.

### Conditions

Measurement views: All

### Group

System commands

### Syntax

```
SYSTem:PRESet:USER:SElected <file_name> SYSTem:PRESet:USER:SElected?
```

### Arguments

<file\_name>::<string> specifies the User presets file to set. The file extension is .Setup. You can omit the extension.

### Examples

SYSTem:PRESet:USER:SElected "Spectrum" sets the User preset to "Spectrum.Setup" saved in C:\RSA5100B Files\User Presets directory.

## SYSTem:PRESet:WLAN:AD:STANdard

Sets or queries the 802.11ad preset standard.

### Conditions

Measurement views: All 802.11ad measurement views

### Group

System commands

### Syntax

```
SYSTem:PRESet:WLAN:AD:STANdard {SC|CPHY,SEM|EVM,0|1,0|1,CH1|CH2}  
SYSTem:PRESet:WLAN:AD:STANdard?
```

### Arguments

SC|CPHY = Sets standard as either Single carrier or Control PHY SEM|EVM= Sets Test setup as either SEM or EVM

0|1= Checks or unchecks the check box for Retain current Center Frequency settings

0|1= Checks or unchecks the check box for Retain Current Ref Level setting CH1|CH2= Selects between channel 1 and channel 2

### Returns

See Arguments



## Examples

`SYSTEM:PRESET:WLAN:AD:STANDARD` might return "SC,EVM,0,1,CH2", indicating the standard selected is single carrier, test setup selected is EVM, Retain current Center Frequency box is disabled (unchecked), Retain Current Ref Level is enabled (checked), and Channel 2 is selected.

## SYSTEM:PRESet:WLAN:AY:STANdard

Set or query the WLAN 802.11ay standard, Retain Current Center Frequency setting, and Retain Current Reference Level setting in the WLAN 802.11ay standard preset.

## Conditions

Measurement views: All WLAN 802.11ay

## Group

System commands

## Syntax

`SYSTEM:PRESet:WLAN:AY:STANdard <AYSC>,{ EVM | SEM },{ 0 | 1},{ 0 | 1},{ CH1 | CH2},{ BW1X | BW2X }`

`SYSTEM:PRESet:WLAN:AY:STANdard?`

## Arguments

Argument	Description
AYSC	Sets the standard to Single Carrier.
EVM   SEM	Sets the test setup as EVM or SEM.
0   1	Retains the Center Frequency.
0   1	Retains the Reference Level.
CH1   CH2	Sets the Radio Frequency Source to Channel 1 or Channel 2.
BW1X   BW2X	Sets the Channel Bandwidth as 2.16 GHz or 4.32 GHz.

## Returns

See Arguments.

## Examples

`SYST:PRE:WLAN:AY:STAN AYSC,EVM,0,1,CH2,BW1X` sets preset to 802.11ay, standard to Single Carrier, disables the Retain Current Center Frequency setting, enables the Retain Current Reference Level setting, and sets the channel to CH2.

`SYST:PRE:WLAN:AY:STAN?` may return `AYSC,SEM,1,0,CH1,BW2X`, which means that the standard is Single Carrier, the display type is SEM, the Center Frequency is retained, the Reference Level is not retained, Reference Frequency Source is set to channel 1, and the Channel Bandwidth is 4.32 GHz.

## SYSTEM:PRESet:WLAN:STANdard

Sets or queries the standard and bandwidth presets for the WLAN standards preset type. The first enumeration is the 802.11 standard; the second enumeration is the bandwidth.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:PRESet:WLAN:STANdard { B | AC, [BW20|BW40|BW80|BW160] | [A,G],BW20 | P, [BW5|BW10|BW20] | J, [BW10|BW20] | N, [BW20|BW40] }
```

```
SYSTem:PRESet:WLAN:STANdard?
```

## Arguments

B selects the 802.11b WLAN standard preset options.

AC, [BW20|BW40|BW80|BW160] selects the 802.11b WLAN standard preset options.

[A,G], BW20 selects the 802.11a or g WLAN standard preset options.

P, [BW5|BW10|BW20] selects the 802.11p WLAN standard preset options.

J, [BW10|BW20] selects the 802.11j WLAN preset options.

N, [BW20|BW40] selects the 802.11n WLAN standard preset options.

## Examples

SYSTEM:PRESET:WLAN:STANDARD P, BW5 selects 802.11p as the WLAN standard to use and 5 MHz as the bandwidth preset.

## SYSTEM:SVPC:DISconnect (No Query Form)

Disconnects an RSA306B/500A/600A/7100A. If playback is active, exits playback.

## Group

System commands

## Syntax

```
SYSTem:SVPC:DISconnect
```

## Arguments

None.

## SYSTEM:SVPC:CONNect (No Query Form)

Connects an RSA306B/500A/600A/7100A.

## Group

System commands

## Syntax

```
SYSTem:SVPC:CONNect
```

## Arguments

None.

## SYSTem:SVPC:CONNect:RSA300 (No Query Form)

Connects an RSA306 instrument.

## Group

System commands

## Syntax

```
SYSTem:SVPC:CONNect:RSA300
```

## Arguments

None.

## SYSTem:SVPC:INSTrument:SERial? (Query Only)

Available when an RSA is connected to SignalVu-PC.

Queries the serial number and model of the connected instrument.

## Group

System commands

## Syntax

```
SYSTem:SVPC:INSTrument:SERial?
```

## Arguments

None.

## Returns

Serial number and model of connected instrument.

This commands will not return the correct instrument serial number if an RSA is disconnected and then another connected without using the Search for Instruments tool in the SignalVu-PC application to find and connect to the new instrument.

## SYSTem:SVPC:INSTrument:VERSion? (Query Only)

Available when an RSA is connected to SignalVu-PC. Queries the firmware version of the connected instrument.

## Group

System commands

## Syntax

`SYSTem:SVPC:INSTRument:VERSion?`

## Arguments

None.

## Returns

Firmware version of connected instrument.

This commands will not return the correct instrument firmware version if an RSA is disconnected and then another connected without using the Search for Instruments tool in the SignalVu-PC application to find and connect to the new instrument.

## **SYSTem:SVPC:INSTRument:MODEL? (Query Only)**

Available when an RSA is connected to SignalVu-PC. Queries the model of the connected instrument.

## Group

System commands

## Syntax

`SYSTem:SVPC:INSTRument:MODEL?`

## Arguments

None.

## Returns

Model of connected instrument.

This commands will not return the correct model if an RSA is disconnected and then another connected without using the Search for Instruments tool in the SignalVu-PC application to find and connect to the new instrument.

## **SYSTem:SVPC:INSTRument:OPTions? (Query Only)**

Available when an RSA is connected to SignalVu-PC. Queries the options for the connected instrument.

## Group

System commands

## Syntax

`SYSTem:SVPC:INSTRument:OPTions?`

## Arguments

None.

## Returns

Model of connected instrument.

This commands will not return the correct options if an RSA is disconnected and then another connected without using the Search for Instruments tool in the SignalVu-PC application to find and connect to the new instrument.

## SYSTem:TIME

Sets or queries the time (hours, minutes, and seconds). This command is equivalent to the time setting through the Windows Control Panel.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

```
SYSTem:TIME <hour>,<minute>,<second> SYSTem:TIME?
```

## Related Commands

[SYSTem:DATE](#)

## Arguments

<hour>: : <NRf> specifies the hours. Range: 0 to 23.

<minute>: : <NRf> specifies the minutes. Range: 0 to 59.

<second>: : <NRf> specifies the seconds. Range: 0 to 59. These values are rounded to the nearest integer.

\*RST has no effect on the settings.

## Examples

```
SYSTEM:TIME10,15,30 sets the time to 1015:30.
```

## SYSTem:VERSion? (Query Only)

Returns the SCPI version number for which the analyzer complies.

## Conditions

Measurement views: All

## Group

System commands

## Syntax

SYSTem:VERSion?

## Arguments

None

## Returns

<NR2> has the form YYYY.V where the Ys represent the year-version (for example, 1999) and the V represents an approved revision number for that year.

## Examples

SYSTEM:VERSION? might return 1999.0 for the SCPI version.

# Trace Commands

## TRACe:{AM|FM|PM}:DETection

Sets or queries the display detector, the method to be used for decimating traces to fit the available horizontal space on screen. The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

## Conditions

Measurement views: AM, FM, PM

## Group

Trace commands

## Syntax

```
TRACe:{AM|FM|PM}:DETection { AVERAge | POSitive | NEGative }
```

```
TRACe:{AM|FM|PM}:DETection?
```

## Arguments

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

## Examples

TRACE:AM:DETECTION AVERAge specifies that the trace displays the average data value for each pixel.

## TRACe:{AM|FM|PM}:FREeze

Determines whether or not to freeze the trace display in the AM/FM/PM view.

## Conditions

Measurement views: AM, FM, PM

## Group

Trace commands

## Syntax

```
TRACe:{AM|FM|PM}:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe:{AM|FM|PM}:FREeze?
```

## Arguments

OFF or 0 updates the trace display normally.

ON or 1 stops updating the trace display.

## Examples

```
TRACE:AM:FREEZE ON freezes the trace display.
```

## TRACe:{AM|FM|PM}:FUNCTion

Sets or queries the trace function in the AM/FM/PM view.

## Conditions

Measurement views: AM, FM, PM

## Group

Trace commands

## Syntax

```
TRACe:{AM|FM|PM}:FUNCTion { NORMal }
```

```
TRACe:{AM|FM|PM}:FUNCTion?
```

## Arguments

NORMal selects the normal display.

## Examples

```
TRACE:AM:FUNCTION NORMal selects the normal display in the AM view.
```

## TRACe:DIQVtime[:MEASview<y>]:ENABLE:I

Enables or disables the I trace in the Demod I&Q versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Trace commands

## Syntax

```
TRACe:DIQVtime[:MEASview<y>]:ENABLE:I { OFF | ON | 0 | 1 }
```

```
TRACe:DIQVtime[:MEASview<y>]:ENABLE:I?
```

## Arguments

OFF or 0 hides the I trace.

ON or 1 shows the I trace.

## Examples

TRACe:DIQVtime:MEASview1:ENABLE:I ON shows the I trace in the Demod I&Q versus Time measurement view 1.

## TRACe:DIQVtime[:MEASview<y>]:ENABLE:Q

Enables or disables the Q trace in the Demod I&Q versus Time measurement.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Trace commands

## Syntax

```
TRACe:DIQVtime[:MEASview<y>]:ENABLE:Q { OFF | ON | 0 | 1 }
```

```
TRACe:DIQVtime[:MEASview<y>]:ENABLE:Q?
```

## Arguments

OFF or 0 hides the Q trace.

ON or 1 shows the Q trace.



## Examples

`TRACe:DIQVtime:MEASview1:ENABle:Q ON` shows the Q trace in the Demod I&Q versus Time Measurement view 1.

## TRACe:DIQVtime[:MEASview<y>]:MODE

Sets or queries whether to display the Demod I&Q versus Time trace as vectors or symbols (points).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement view: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Trace commands

### Syntax

```
TRACe:DIQVtime[:MEASview<y>]:MODE { VECTors | SYMBols }
```

### Arguments

`VECTors` displays the trace in a format that uses lines to connecting points in the trace display.

`SYMBols` displays the trace in a format that does not use lines to connect points in the trace display.

## Examples

`TRACe:DIQVtime:MEASview1:MODE SYMBols` sets the display to show symbols without lines connecting points Measurement view 1.

## TRACe:DIQVtime[:MEASview<y>]:SElect:I

Selects the I trace in the Demod I&Q versus Time. The query version of this command returns whether the I trace is selected or not.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

### Group

Trace commands

## Syntax

```
TRACe:DIQVtime[:MEASview<y>]:SElect:I
```

```
TRACe:DIQVtime[:MEASview<y>]:SElect:I?
```

## Arguments

None

## Returns

0 indicates that the I trace is deselected.

1 indicates that the I trace is selected.

## Examples

TRACe:DIQVtime:MEASview1:SElect:I selects the I trace in the Demod I&Q versus Time Measurement view 1.

## TRACe:DIQVtime[:MEASview<y>]:SElect:Q

Selects the Q trace in the Demod I&Q versus Time. The query version of this command returns whether the Q trace is selected or not.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Demod I&Q versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod I&Q versus Time).

## Group

Trace commands

## Syntax

```
TRACe:DIQVtime[:MEASview<y>]:SElect:Q
```

```
TRACe:DIQVtime[:MEASview<y>]:SElect:Q?
```

## Arguments

None

## Returns

0 indicates that the Q trace is deselected.

1 indicates that the Q trace is selected.

## Examples

TRACe:DIQVtime[:MEASview1]:SElect:Q selects the Q trace in the Demod I&Q versus Time Measurement view 1.

## TRACe:EDiagram[:MEASview<y>]:ENABLE:I

Determines whether to show or hide the I trace in the Eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

### Group

Trace commands

### Syntax

```
TRACe:EDiagram[:MEASview<y>]:ENABLE:I { OFF | ON | 0 | 1 }
```

```
TRACe:EDiagram[:MEASview<y>]:ENABLE:I?
```

### Arguments

OFF or 0 hides the I trace.

ON or 1 shows the I trace.

### Examples

TRACe:EDiagram:MEASview2:ENABLE:I ON shows the I trace in the eye diagram Measview 2.

## TRACe:EDiagram[:MEASview<y>]:ENABLE:Q

Determines whether to show or hide the Q trace in the Eye diagram.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

### Group

Trace commands

### Syntax

```
TRACe:EDiagram[:MEASview<y>]:ENABLE:Q { OFF | ON | 0 | 1 }
```

```
TRACe:EDiagram[:MEASview<y>]:ENABLE:Q?
```

## Arguments

OFF or 0 hides the Q trace.

ON or 1 shows the Q trace.

## Examples

`TRACe:EDIagram:MEASview2:ENABle:Q ON` shows the Q trace in the eye diagram Measview 2.

## TRACe:EDIagram[:MEASview<y>]:SElect:I

Selects the I trace in the Eye diagram. When used as query, it returns whether the I trace is selected or not.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Trace commands

## Syntax

```
TRACe:EDIagram[:MEASview<y>]:SElect:I
```

```
TRACe:EDIagram[:MEASview<y>]:SElect:I?
```

## Arguments

None

## Returns

0 indicates that the I trace is deselected.

1 indicates that the I trace is selected.

## Examples

`TRACe:EDIagram:MEASview2:SElect:I` selects the I trace in the eye diagram for measview2.

## TRACe:EDIagram[:MEASview<y>]:SElect:Q

Selects the Q trace in the eye diagram. The query version of this command returns whether the Q trace is selected or not.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Eye diagram

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: Eye diagram).

## Group

Trace commands

## Syntax

```
TRACe:EDIagram[:MEASview<y>]:SElect:Q
```

```
TRACe:EDIagram[:MEASview<y>]:SElect:Q?
```

## Arguments

None

## Returns

0 indicates that the Q trace is deselected.

1 indicates that the Q trace is selected.

## Examples

TRACe:EDIagram:MEASview2:SElect:Q selects the Q trace in the eye diagram Measview 2.

## TRACe:EMC:AMBIent:COUNT

Sets or queries the count value of specified ambient trace in EMC view.

<value> = <NR1> count value

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:AMBIent:COUNT <value>
```

```
TRACe:EMC:AMBIent:COUNT?
```

## Arguments

<value> = <NR1>

## Returns

<NR1>

## Examples

`TRACE:EMC:AMBIENT:COUNT` might return 20, indicating that the count value of ambient trace 2 is 20.

## TRACe:EMC:AMBIent:COUNT:ENABLE

Sets or queries to enable or disable the count check box for specified ambient trace in EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Trace commands

### Syntax

```
TRACe:EMC:AMBIent:COUNT:ENABLE {0 | OFF | 1 | ON}
```

```
TRACe:EMC:AMBIent:COUNT:ENABLE?
```

### Arguments

0 or OFF disables the count check box 1 or ON enables the count check box

### Returns

<Boolean> 0, 1

## Examples

`TRACE:EMC:AMBIENT:COUNT:ENABLE` might return 1, indicating that count check box is enabled.

## TRACe:EMC:AMBIent:COUNT:RESet (No Query Form)

Resets the count value of live ambient trace in EMC view.

### Conditions

Measurement view: EMC - EMI Display

### Group

Trace commands

### Syntax

```
TRACe:EMC:AMBIent:COUNT:RESet
```

### Arguments

None

## Returns

None

## Examples

`TRACE:EMC:AMBIENT:COUNT:RESET` resets the count value of live ambient trace

## TRACe:EMC:AMBIent:ENABLE

Sets or queries to show or hide the ambient trace on plot by enabling or disabling the show check box in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

`TRACe:EMC:AMBIent:ENABle {0 | OFF | 1 | ON}`

`TRACe:EMC:AMBIent:ENABle?`

## Arguments

0 or OFF disables the show check box and hides the ambient trace 1 or ON enables the show check box and displays the ambient trace

## Returns

<Boolean> 0, 1

## Examples

`TRACE:EMC:AMBIENT:ENABLE?` might returns 0, indicating that the selected ambient trace is hidden.

## TRACe:EMC:AMBIent:FREeze

Sets or queries to freeze the ambient trace on plot by enabling or disabling the freeze check box in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

`TRACe:EMC:AMBIent:FREeze {0 | OFF | 1 | ON}`

`TRACe:EMC:AMBIent:FREeze?`

## Arguments

0 or OFF disables the freeze check box 1 or ON enables the freeze check box

## Returns

<Boolean> 0, 1

## Examples

TRACE:EMC:AMBIENT:FREEZE? might return 1, indicating that the selected ambient trace is frozen.

## TRACe:EMC:AMBIent:FUNCtion

Sets or queries the scan function for the selected ambient trace in EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:AMBIent:FUNCtion {NONE | MAXH | AVER | AVGL}
```

```
TRACe:EMC:AMBIent:FUNCtion?
```

## Arguments

NONE = Normal is the function selected MAXH = Max Hold is the function selected AVER = Avg(VRMS) is the function selected AVGL = Avg (of logs) is the function selected

## Returns

<String> See Arguments

## Examples

TRACE:EMC:AMBIENT:FUNCTION might return AVER, indicating the scan function selected for ambient trace is Avg(VRMS).

## TRACe:EMC:AMBIent:SElect

Sets or queries the selection of ambient trace in live trace of EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands



## Syntax

```
TRACe:EMC:AMBIent:SElect
```

```
TRACe:EMC:AMBIent:SElect?
```

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

TRACE:EMC:AMBIENT:SELECT might return 1, indicating that ambient trace 1 is selected in live trace.

## TRACe:EMC:MATH:ENABLE

Sets or queries to show or hide the math trace on plot by enabling or disabling the show check box in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:MATH:ENABle {0 | OFF | 1 | ON}
```

```
TRACe:EMC:MATH:ENABle?
```

## Arguments

0 or OFF disables the show check box and hides the math trace 1 or ON enables the show check box and displays the math trace

## Returns

<Boolean> 0, 1

## Examples

TRACE:EMC:MATH:ENABLE? might return 1, indicating that the math trace is enabled (displayed).

## TRACe:EMC:MATH:FREeze

Sets or queries to freeze the math trace on plot by enabling or disabling the freeze check box in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:MATH:FREeze {0 | OFF | 1 | ON}
```

```
TRACe:EMC:MATH:FREeze?
```

## Arguments

0 or OFF disables the freeze check box 1 or ON enables the freeze check box

## Returns

<Boolean> 0, 1

## Examples

TRACE:EMC:MATH:FREEZE? might return 1, indicating that selected math trace is frozen.

## TRACe:EMC:MATH:LEFToperand

Sets or queries the trace for left operand to obtain a math trace in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:MATH:LEFToperand <arg>
```

```
TRACe:EMC:MATH:LEFToperand?
```

## Arguments

<arg> = {scan1 | scan2 | ambient | recall1 | recall2 | recall3 | recall4 | recall5}

## Returns

<String>

## Examples

TRACE:EMC:MATH:LEFTOPERAND selects ambient trace as left operand.

## TRACe:EMC:MATH:RIGHToperand

Sets or queries the trace for right operand to obtain a math trace in EMC view.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:MATH:RIGHToperand <arg>
```

```
TRACe:EMC:MATH:RIGHToperand?
```

## Arguments

<arg> = {scan1 | scan2 | ambient | recall1 | recall2 | recall3 | recall4 | recall5}

## Returns

<String>

## Examples

TRACe:EMC:MATH:RIGHTOPERAND? might return scan1, indicating that the right operand selected is scan1 trace.

## TRACe:EMC:MATH:SElect

Sets or queries the selection of math trace in live trace of EMC View.

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe:EMC:MATH:SElect
```

```
TRACe:EMC:MATH:SElect?
```

## Arguments

None

## Returns

<Boolean> 0, 1

## Examples

TRACe:EMC:MATH:SELECT might return 1, indicating that math trace is selected.

## TRACe:EVM[:MEASview<y>]:MODE

Sets or queries whether to display the EVM versus Time trace as vectors or symbols (points).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement view: EVM versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: EVM versus Time).

### Group

Trace commands

### Syntax

```
TRACe:EVM[:MEASview<y>]:MODE { VECTors | SYMBols }
```

```
TRACe:EVM[:MEASVIEW<y>]:MODE?
```

### Arguments

VECTors displays the trace in a format that uses lines to connecting points in the trace display.

SYMBols displays the trace in a format that does not use lines to connect points in the trace display.

### Examples

TRACe:EVM:MEASVIEW2:MODE VECTors sets the display to connect points in the display for measview 2 of EVM versus Time.

## TRACe:FDVTime[:MEASview<y>]:MODE

Sets or queries whether to display the Frequency Deviation versus Time trace as vectors or symbols (points).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement view: Frequency deviation versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Frequency deviation versus Time).

### Group

Trace commands

### Syntax

```
TRACe:FDVTime[:MEASview<y>]::MODE { VECTors | SYMBols }
```

```
TRACe:FDVTime[:MEASView<y>]:MODE?
```

## Arguments

`VECTors` display the trace in a format that uses lines to connect points in the trace display.

`SYMBOLs` display the trace in a format that does not use lines to connect points in the trace display.

## Examples

`TRACE:FDVTime:MEASview2:MODE VECTORS` sets the display to connect points in the display for measview 2 of Frequency deviation versus Time.

## TRACe:{FSETtling|PSETtling}:AVERAge:COUNT

Sets or queries the number of traces to combine. This command is effective when you select `AVERAge` with the `s` command.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Trace commands

## Syntax

`TRACe:{FSETtling|PSETtling}:AVERAge:COUNT <number>`

`TRACe:{FSETtling|PSETtling}:AVERAge:COUNT?`

## Related Commands

`TRACe:{FSETtling|PSETtling}:AVERAge:ENABLE`

## Arguments

`<number>::<NR1>` specifies the number of traces to combine for averaging.

Range: 1 to 10000.

## Examples

`TRACE1:PSETTLING:AVERAGE:COUNT 64` sets the average count to 64 for Trace 1.

## TRACe:{FSETtling|PSETtling}:AVERAge:COUNT:CURRent? (Query Only)

Queries the current running average count value. The returned value is valid (non-zero) only while the instrument is acquiring and averaging a signal. Use the `TRACe:{FSETtling|PSETtling}:AVERAge:COUNT` command to determine the setting for Average.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Trace commands

## Syntax

```
TRACe:{FSETtling|PSETtling}:AVERAge:COUNT:CURRent?
```

## Related Commands

```
TRACe:{FSETtling|PSETtling}:AVERAge:COUNT
```

## Examples

TRACE1:PSETTLING:AVERAGE:COUNT? might return the value 64.

## TRACe:{FSETtling|PSETtling}:AVERAge:ENABLE

Determines whether to enable or disable averaging the trace(s).

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Trace commands

## Syntax

```
TRACe:{FSETtling|PSETtling}:AVERAge:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe:{FSETtling|PSETtling}:AVERAge:ENABLE?
```

## Arguments

OFF or 0 disables averaging.

ON or 1 enables averaging.

## Examples

TRACE:PSETTLING:AVERAGE:ENABLE ON enables the Phase Settling Time averaging.

## TRACe:{FSETtling|PSETtling}:RESet (No Query Form)

If trace averaging is enabled, this command resets the current averaged trace to 0.

## Conditions

Measurement views: Frequency and Phase Settling Time

## Group

Trace commands

## Syntax

```
TRACe:{FSETtling|PSETtling}:RESet
```

## Examples

TRACE1:PSETTLING:RESET resets the current averaged trace to 0.

## TRACe:{FSETtling|PSETtling}:SMOothing:COUNT

Sets or queries the number of data points to take the moving average for smoothing the traces. This command applies to both Trace 1 and Trace 2.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Trace commands

### Syntax

```
TRACe:{FSETtling|PSETtling}:SMOothing:COUNT <number>
```

```
TRACe:{FSETtling|PSETtling}:SMOothing:COUNT?
```

### Arguments

<number>::<NR1> specifies the number of data points to take the moving average for smoothing. Range: 2 to 1000.

## Examples

TRACE:FSETTLING:SMOOTHING:COUNT 16 sets the Frequency Settling Time smoothing count to 16.

## TRACe:{FSETtling|PSETtling}:SMOothing:ENABLE

Determines whether to enable or disable smoothing the trace(s).

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Trace commands

### Syntax

```
TRACe:{FSETtling|PSETtling}:SMOothing:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe:{FSETtling|PSETtling}:SMOothing:ENABLE?
```

### Arguments

OFF or 0 disables smoothing.

ON or 1 enables smoothing.

## Examples

TRACE:PSETTLING:SMOOTHING:ENABLE ON enables the Phase Settling Time smoothing.

## TRACe<x>:FVTime[:MEASview<y>]

Determines whether or not to show the trace in the Frequency versus Time view.

### Conditions

Measurement views: Frequency versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:FVTime[:MEASview<y>] { OFF | ON | 0 | 1 }
```

```
TRACe<x>:FVTime[:MEASview<y>]?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the trace in the Frequency versus Time view.

ON or 1 shows the trace in the Frequency versus Time view.

### Examples

TRACE1:FVTIME:MEASview1 ON shows the trace in the Frequency versus Time view.

## TRACe<x>:FVTime[:MEASview<y>]:AVERage:COUNT

Sets or queries the number of traces to combine. This command is effective when you select AVERage with the TRACe:FVTime[:MEASview<y>]:FUNCTION command.

### Conditions

Measurement views: Frequency versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:AVERage:COUNT <number>
```

```
TRACe<x>:FVTime[:MEASview<y>]:AVERage:COUNT?
```



## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<number> : : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

## Examples

TRACe1:FVTime:MEASview1:AVERAGE:COUNT 64 sets the average count to 64.

## TRACe<x>:FVTime[:MEASview<y>]:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Frequency versus Time measurement. This command is effective when TRACe<x>:FVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

TRACe<x>:FVTime[:MEASview<y>]:COUNT <number>

TRACe<x>:FVTime[:MEASview<y>]:COUNT?

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<number> : : <NR1> specifies the count for Max/Min Hold.

Range: 1 to 10000.

## Examples

TRACe1:FVTime:MEASview1:COUNT 32 sets the count to 32 for the Max/Min Hold trace.

## TRACe<x>:FVTime[:MEASview<y>]:COUNT:ENABLE

Determines whether to enable or disable the count for the Max or Min Hold trace in the Frequency versus Time view. This command is effective when TRACe<x>:FVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:FVTime[:MEASview<y>]:COUNT:ENABLE?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables the count for the Max/Min Hold trace.

ON or 1 enables the count for the Max/Min Hold trace.

## Examples

```
TRACe1:FVTime:MEASview1:COUNT:ENABLE ON enables the Max/Min Hold count.
```

## TRACe<x>:FVTime[:MEASview<y>]:COUNT:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the Frequency versus Time view. This command is effective when TRACe<x>:FVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:COUNT:RESet
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

```
TRACe1:FVTime:MEASview1:COUNT:RESet clears the Max/Min Hold data and counter, and restarts the process.
```

## TRACe<x>:FVTime[:MEASview<y>]:FREeze

Determines whether or not to freeze the display of the trace in the Frequency versus Time measurement.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:FVTime[:MEASview<y>]:FREeze?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 updates the display of the trace normally.

ON or 1 stops updating the display of the trace.

## Examples

```
TRACE1:FVTIME:MEASview1:FREEZE ON stops updating the display of the trace.
```

## TRACe<x>:FVTime[:MEASview<y>]:DETection

Enables or queries the type of detection for the specified trace. The parameter

<x> represents traces 1 to 4.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative | POSNegative | SAMPLE }
```

```
TRACe<x>:FVTime[:MEASview<y>]:DETection?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

`NEGative` displays the minimum data value for each pixel.

`POSNegative` displays the maximum and minimum data values for each pixel.

`SAMPle` displays the first sample value received for each pixel.

## Examples

`TRACe2:FVTime[:MEASview<y>]:DETEction POSitive` enables positive detection on Trace2.

## TRACe<x>:FVTime[:MEASview<y>]:FUNCTION

Sets or queries the trace function in the Frequency versus Time measurement.

### Conditions

Measurement views: Frequency versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:FUNCTION { NORMal | AVERage | MAXHold | MINHold}
```

```
TRACe<x>:FVTime[:MEASview<y>]:FUNCTION?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

`NORMal` selects the normal waveform display.

`AVERage` selects the Average display that indicates the average frequency drift at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum frequency drift at each time point.

`MINHold` selects the Min Hold display that indicates the minimum frequency drift at each time point.

### Examples

`TRACe1:FVTIME:MEASview1:FUNCTIONMAX Hold` displays the Max Hold trace in the Frequency versus Time measurement.

## TRACe<x>:IQVTime[:MEASview<y>]:AVERage:COUNT

Sets or queries the number of traces to combine. This command works for both I and Q traces when you select AVERage in the `TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION` command.

### Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:AVERAge:COUNT <number>
```

```
TRACe<x>:IQVTime[:MEASview<y>]:AVERAge:COUNT?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<number>: : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

## Examples

TRACE1:IQVTIME:MEASview1:AVERAGE:COUNT 64 sets the average count to 64.

## TRACe<x>:IQVTime[:MEASview<y>]:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the RF I & Q versus Time measurement. This command is effective when TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

## Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:COUNT <number>
```

```
TRACe<x>:IQVTime[:MEASview<y>]:COUNT?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<number>: : <NR1> specifies the count for Max/Min Hold.

Range: 1 to 10000.

## Examples

TRACE1:IQVTIME:MEASview1:COUNT 32 sets the count to 32 for the Max/Min Hold trace.

## TRACe<x>:IQVTime[:MEASview<y>]:COUNT:ENABLE

Determines whether to enable or disable the count for the Max or Min Hold trace in the RF I & Q versus Time view. This command is effective when TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:COUNT:ENABLE?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables the count for the Max/Min Hold trace.

ON or 1 enables the count for the Max/Min Hold trace.

### Examples

TRACE1:IQVTIME:MEASview1:COUNT:ENABLE ON enables the Max/Min Hold count.

## TRACe<x>:IQVTime[:MEASview<y>]:COUNT:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the RF I & Q versus Time view. This command is effective when TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:COUNT:RESet
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

TRACE1:IQVTIME:MEASview1:COUNT:RESET clears the Max/Min Hold data and counter, and restarts the process.

## TRACe<x>:IQVTime[:MEASview<y>]:DETection

Sets or queries the detection method for the RF I & Q versus Time view.

## Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative |
POSNegative | SAMPlE }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:DETection?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

## Examples

TRACE1:IQVTIME:MEASview1:DETECTION AVERAGE enables display of the average value for each pixel.

## TRACe<x>:IQVTime[:MEASview<y>]:ENABLe:I

Determines whether to show or hide the I trace in the RF I & Q versus Time measurement.

## Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:ENABLE:I { OFF | ON | 0 | 1 }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:ENABLE:I?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the I trace.

ON or 1 shows the I trace.

## Examples

```
TRACE1:IQVTIME:MEASview1:ENABLE:I ON shows the I trace in the RF I & Q versus Time measurement.
```

## TRACe<x>:IQVTime[:MEASview<y>]:ENABLE:Q

Determines whether to show or hide the Q trace in the RF I & Q versus Time measurement.

## Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:ENABLE:Q { OFF | ON | 0 | 1 }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:ENABLE:Q?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 hides the Q trace.

ON or 1 shows the Q trace.

## Examples

```
TRACE1:IQVTIME:MEASview1:ENABLE:Q ON shows the Q trace in the IQ level versus Time measurement.
```



## TRACe<x>:IQVTime[:MEASview<y>]:FREeze

Determines whether to freeze the IQ traces in the RF I & Q versus Time measurement.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:FREeze?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 updates IQ trace display normally.

ON or 1 stops updating IQ trace display.

### Examples

```
TRACE1:IQVTIME:MEASview1:FREEZE ON freezes the IQ traces.
```

## TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION

Sets or queries the trace function in the RF I & Q versus Time measurement.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION { NORMa1 | AVERage | MAXHold | MINHold }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:FUNCTION?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

NORMa1 selects the normal waveform display.

**AVERage** selects the Average display that indicates the average signal level at each time point.

**MAXHold** selects the Max Hold display that indicates the maximum signal level at each time point.

**MINHold** selects the Min Hold display that indicates the minimum signal level at each time point.

## Examples

`TRACE1:IQVTIME:MEASview1:FUNCTION MAXHold` displays the Max Hold trace in the IQ level versus Time measurement.

## TRACe<x>:IQVTime[:MEASview<y>]:SElect:I

Determines whether or not to select the I trace to obtain the maximum and minimum measurement results.

### Conditions

Measurement views: RF I & Q versus Time

### Group

Trace commands

### Syntax

`TRACe<x>:IQVTime[:MEASview<y>]:SElect:I { OFF | ON | 0 | 1 }`

`TRACe<x>:IQVTime[:MEASview<y>]:SElect:I?`

### Related Commands

`TRACe<x>:IQVTime[:MEASview<y>]:SElect:Q`

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional.  
Default value 1.

OFF or 0 deselects the I trace.

ON or 1 selects the I trace.

Executing `TRACe<x>:IQVTime[:MEASview<y>]:SElect:I ON` sets

`TRACe<x>:IQVTime[:MEASview<y>]:SElect:Q OFF`.

### Examples

`TRACE1:IQVTIME:MEASview1:SELECT:I ON` selects the I trace in the RF I & Q versus Time measurement.

## TRACe<x>:IQVTime[:MEASview<y>]:SElect:Q

Determines whether or not to select the Q trace to obtain the maximum and minimum measurement results.

### Conditions

Measurement views: RF I & Q versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:IQVTime[:MEASview<y>]:SElect:Q { OFF | ON | 0 | 1 }
```

```
TRACe<x>:IQVTime[:MEASview<y>]:SElect:Q?
```

## Related Commands

```
TRACe<x>:IQVTime[:MEASview<y>]:SElect:I
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 deselects the Q trace.

ON or 1 selects the Q trace.

Executing TRACe<X>:IQVTime[:MEASview<y>]:SElect:Q ON sets  
TRACe<x>:IQVTime[:MEASview<y>]:SElect:I OFF.

## Examples

TRACE1:IQVTIME:MEASview1:SELECT:Q ON selects the Q trace in the RF I&Q versus Time measurement.

## TRACe:MERRor[:MEASview<y>]:MODE

Sets or queries whether to display the Magnitude error trace as vectors or symbols (points).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement view: Magnitude error

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Magnitude error versus Time).

## Group

Trace commands

## Syntax

```
TRACe:MERRor[:MEASview<y>]:MODE { VECTors | SYMBols }
```

```
TRACe:MERRor[:MEASview<y>]:MODE?
```

## Arguments

VECTors displays the trace in a format that uses lines to connecting points in the trace display.

`SYMBOLs` displays the trace in a format that does not use lines to connect points in the trace display.

## Examples

`TRACE:MERRor:MEASview2:MODE VECTORS` sets the display to connect points in the display for measview 2 of MERRor versus Time.

## TRACe:OBW:MAXHold

Determines whether or not to perform a Max Hold on the spectrum data for the Occupied Bandwidth trace.

### Conditions

Measurement views: Occupied Bandwidth

### Group

Trace commands

### Syntax

```
TRACe:OBW:MAXHold { OFF | ON | 0 | 1 }
```

```
TRACe:OBW:MAXHold?
```

### Arguments

OFF or 0 does not perform a Max Hold on the spectrum data.

ON or 1 performs a Max Hold on the spectrum data.

## Examples

`TRACE:OBW:MAXHOLDON` performs a Max Hold on the spectrum data for the Occupied Bandwidth trace.

## TRACe:OFDM:CONSte

Determines whether or not to show the trace in the OFDM Constellation view.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe:OFDM:CONSte { OFF | ON | 0 | 1 }
```

```
TRACe:OFDM:CONSte?
```

### Arguments

OFF or 0 hides the trace in the view.

ON or 1 shows the trace in the view.

## Examples

`TRACe:OFDM:CONStE ON` shows the trace in the OFDM Constellation view

## TRACe:OFDM:CONStE:FR EEze

Determines whether or not to freeze the display of the trace in the OFDM Constellation view.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe:OFDM:CONStE:FR EEze { OFF | ON | 0 | 1 }
```

```
TRACe:OFDM:CONStE:FR EEze?
```

### Arguments

OFF or 0 updates the display of the trace normally.

ON or 1 stops updating the display of the trace.

## Examples

`TRACe:OFDM:CONStE:FR EEZE ON` freezes the trace in the OFDM Constellation view.

## TRACe:OFDM:FLATness:FR EEze

Sets or queries whether to freeze the average trace on the OFDM Spectral Flatness display. When the freeze state is turned on, the trace stops updating.

### Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

### Group

Trace commands

### Syntax

```
TRACe:OFDM:FLATness:FR EEze { 0 | 1 | OFF | ON }
```

```
TRACe:OFDM:FLATness:FR EEze?
```

## Arguments

ON or 1 turns on the freeze state, causing the trace to stop updating.

OFF or 0 turns off the freeze state.

## Returns

Outputs the frozen status of the selected trace as an integral value. 1 indicates a frozen trace and 0 a non-frozen trace.

## Examples

TRAC : OFDM : FLAT : FRE 1 freezes the trace, causing it to stop updating.

## TRACe:OFDM:FLATness:SHOW

Displays or hides the trace on the OFDM Spectral Flatness display.

## Conditions

Measurement view: OFDM

This command requires Option 22, "OFDM Measurements".

## Group

Trace commands

## Syntax

```
TRACe:OFDM:FLATness:SHOW {0|1|OFF|ON}
```

## Arguments

ON or 1 displays the trace.

OFF or 0 hides the trace.

## Examples

TRACe:OFDM:FLAT:SHOW 1 turns on the display of the trace.

## TRACe:PERRor[:MEASview<y>]:MODE

Sets or queries whether to display the Frequency deviation versus Time trace as vectors or symbols (points).

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement view: Phase error versus Time

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Phase error versus Time).

## Group

Trace commands

## Syntax

```
TRACe:PERRor[:MEASview<y>]:MODE { VECTors | SYMBols }
```

```
TRACe:PERRor[:MEASview<y>]:MODE?
```

## Arguments

**VECTors** displays the trace in a format that uses lines to connecting points in the trace display.

**SYMBols** displays the trace in a format that does not use lines to connect points in the trace display.

## Examples

`TRACe:PERRor:MEASview2:MODE VECTors` sets the display to connect points in the display display for measview 2 of Phase error versus Time .

## TRACe<x>:PHVTime[:MEASview<y>]

Determines whether or not to show the trace in the Phase versus Time view.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>] { OFF | ON | 0 | 1 }
```

```
TRACe<x>:PHVTime[:MEASview<y>]?
```

## Arguments

**<x>** = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

**<y>** = 1 to 8; represents measurement index based on number of sources of connected instrument, where **MEASview** keyword is optional. Default value 1.

**OFF** or **0** hides the trace in the Phase versus Time view.

**ON** or **1** shows the trace in the Phase versus Time view.

## Examples

`TRACE1:PHVTIME:MEASview1 ON` shows the trace in the Phase versus Time view.

## TRACe<x>:PHVTime[:MEASview<y>]:AVERAge:COUNT

Sets or queries the number of traces to combine. This command is effective when you select `AVERAge` with the `TRACe<x>:PHVTime[:MEASview<y>]:FUNctIon` command.

### Conditions

Measurement views: Phase versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:AVERAge:COUNT <number>
```

```
TRACe<x>:PHVTime[:MEASview<y>]:AVERAge:COUNT?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional. Default value 1.

<number>: : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

### Examples

`TRACE1:PHVTIME:MEASview1:AVERAGE:COUNT 64` sets the average count to 64.

## TRACe<x>:PHVTime[:MEASview<y>]:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Phase versus Time measurement. This command is effective when `TRACe<x>:PHVTime[:MEASview<y>]:FUNctIon` is set to `MAXHold` or `MINHold` and `INITiate:CONTinuous` is set to `OFF`.

### Conditions

Measurement views: Phase versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:COUNT <number>
```

```
TRACe<x>:PHVTime[:MEASview<y>]:COUNT?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.



<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

<number>: : <NR1> specifies the count for Max/Min Hold.

Range: 1 to 10000.

## Examples

TRACE1:PHVTIME:MEASview1:COUNT 32 sets the count to 32 for the Max/Min Hold trace.

## TRACe<x>:PHVTime[:MEASview<y>]:COUNT:ENABLE

Determines whether to enable or disable the count for the Max or Min Hold trace in the Phase versus Time view. This command is effective when TRACe<x>:PHVTime:MEASview<y>:FUNCTION is set to MAXHold or MINHold.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:PHVTime[:MEASview<y>]:COUNT:ENABLE?
```

## Related Commands

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 disables the count for the Max/Min Hold trace.

ON or 1 enables the count for the Max/Min Hold trace.

## Examples

TRACE1:PHVTIME:MEASview1:COUNT:ENABLE ON enables the Max/Min Hold count.

## TRACe<x>:PHVTime[:MEASview<y>]:COUNT:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process in the Phase versus Time view. This command is effective when TRACe<x>:PHVTime:MEASview<y>:FUNCTION is set to MAXHold or MINHold.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:COUNT:RESet
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

## Examples

TRACE1:PHVTIME:MEASview1:COUNT:RESET clears the Max/Min Hold data and counter, and restarts the process.

## TRACe<x>:PHVTime[:MEASview<y>]:FREEze

Determines whether to freeze the trace display in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:FREEze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:PHVTime[:MEASview<y>]:FREEze?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

OFF or 0 updates the trace display normally.

ON or 1 stops updating trace display.

## Examples

TRACE1:PHVTIME:MEASview1:FREEZE ON stops updating trace display.

## TRACe<x>:PHVTime[:MEASview<y>]:DETection

Sets or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative |
POSNegative | SAMPlE }
```

```
TRACe<x>:PHVTime[:MEASview<y>]:DETection?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

## Examples

```
TRACe1:PHVTime[:MEASview<y>]:DETection POSitive displays the maximum data value for each pixel.
```

## TRACe<x>:PHVTime[:MEASview<y>]:FUNctioN

Sets or queries the trace function in the Phase versus Time measurement.

## Conditions

Measurement views: Phase versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:FUNctioN { NORMal | AVERAge | MAXHold | MINHold }
```

```
TRACe<x>:PHVTime[:MEASview<y>]:FUNctioN?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

`NORMal` selects the normal waveform display.

`AVERage` selects the Average display that indicates the average phase drift at each time point.

`MAXHold` selects the Max Hold display that indicates the maximum phase drift at each time point.

`MINHold` selects the Min Hold display that indicates the minimum phase drift at each time point.

`POSnegative` displays the maximum and minimum data values for each pixel.

`SAMPle` displays the first sample value received for each pixel.

## Examples

`TRACE1:PHVTIME:MEASview1:FUNCTION MAXHold` displays the Max Hold trace in the Phase versus Time measurement.

## TRACe:SEM:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for multi-trace functions (Max Hold and Average) in the Spectral Emissions Mask measurement. This command is effective when `TRACe:SEM:FUNCTION` is set to `MAXHold` or `AVERage` and `INITiate:CONTinuous` is set to `OFF`.

### Conditions

Measurement views: Spectral Emissions Mask

### Group

Trace commands

### Syntax

`TRACe:SEM:COUNT <number>`

`TRACe:SEM:COUNT?`

### Arguments

`<number>::<NR1>` specifies the count for multi-trace functions.

Range: 1 to 10000.

### Examples

`TRACE:SEM:COUNT 32` sets the count to 32 for multi-trace functions.

## TRACe:SEM:COUNT:RESet (No Query Form)

Clears the multi-function (Max Hold or Average) data and counter and restarts the process in the Spectral Emissions Mask view. This command is effective when `TRACe:SEM:FUNCTION` is set to `MAXHold` or `AVERage`.

### Conditions

Measurement views: Spectral Emissions Mask

## Group

Trace commands

## Syntax

```
TRACe:SEM:COUNT:RESet
```

## Arguments

None

## Examples

TRACe:SEM:COUNT:RESEt clears the multi-function data and counter, and restarts the process.

## TRACe:SEM:FUNction

Sets or queries the trace function in the Spectral Emissions Mask view.

## Conditions

Measurement views: Spectral Emissions Mask

## Group

Trace commands

## Syntax

```
TRACe:SEM:FUNction { NONE | MAXHold | AVERage }
```

```
TRACe:SEM:FUNction?
```

## Arguments

NONE selects normal display.

MAXHold selects the Max Hold display that indicates the maximum amplitude drift at each frequency point.

AVERage selects the Average display that indicates the average amplitude drift at each frequency point.

## Examples

TRACe:SEM:FUNction MAXHold displays the Max Hold trace in the Spectral Emissions Mask measurement.

## TRACe:SGRam[:MEASview<y>]:DETection

Sets or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

## Conditions

Measurement views: Spectrogram

## Group

Trace commands

## Syntax

```
TRACe:SGRam[:MEASview<y>]:DETection { AVERage | POSitive | NEGative | CAVerage |  
CPEak | QUASipeak | AVGLog }
```

```
TRACe:SGRam[:MEASview<y>]:DETection?
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

AVERage displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

CAVerage displays the CISPR average value for each pixel.

CPEak displays the CISPR peak value for each pixel.

QUASipeak displays the quasi-peak value for each pixel.

AVGLog displays the average data value of logs for each pixel.

## Returns

```
{AVERage | POSitive | NEGative | CAVerage | CPEak | QUASipeak | AVGLog}
```

## Examples

```
TRACe:SGRam:MEASview1:DETection POSitive displays the maximum data value for each pixel.
```

## TRACe:SGRam[:MEASview<y>]:FREeze

Determines whether or not to freeze the spectrogram display.

## Conditions

Measurement views: Spectrogram

## Group

Trace commands

## Syntax

```
TRACe:SGRam[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
```

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 updates the display of the spectrogram normally.

ON or 1 stops updating the display of the spectrogram.

## Examples

`TRACe:SGRam:MEASview1:FREeze ON` freezes the spectrogram display.

## TRACe:SGRam[:MEASview<y>]:FUNCTION



**Note:** This function is no longer available in software versions 2.3.0159 and later. Sending this command selects the correct. `SENSe:SGRam[:MEASview<y>]:TIME[:SCALE]:MODE` and `TRACe:SGRam[:MEASview<y>]:DETection` to achieve the same result as with previous software versions. The query will return the equivalent trace function based on the `MODE` and `DETection` settings.

Sets or queries the trace function for the specified trace in the spectrogram.

## Conditions

Measurement views: Spectrogram

## Group

Trace commands

## Syntax

`TRACe:SGRam[:MEASview<y>]:FUNCTION { NONE | AVERage | MAXHold | MINHold| AVGLog }`

`TRACe:SGRam[:MEASview<y>]:FUNCTION?`

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

`NONE` selects the normal spectrogram display.

`AVERage` selects the Average display that indicates the average signal level at each frequency point.

`MAXHold` selects the Max Hold display that indicates the maximum signal level at each frequency point.

`MINHold` selects the Min Hold display that indicates the minimum signal level at each frequency point.

`AVGLog` selects the Average of logs display that indicates the average signal level of logs at each frequency point.

## Returns

{`NONE` | `AVERage` | `MAXHold` | `MINHold` | `AVGLog`}

## Examples

`TRACe:SGRam:MEASview1:FUNCTION MAXHold` selects the Max Hold display for the spectrogram.

## TRACe:SGRam[:MEASview<y>]:FUNCTION:TIME



**Note:** This command has a new function in software versions 2.3.159 and above. Sets the Spectrum Monitor mode values for Time per update. Sending this command sets `SENSe:SGRam[:MEASview<y>]:TIME:SCALE:PER:UPDATE:MINutes` and `SENSe:SGRam[:MEASview<y>]:WATERfall:DIRection` based on the value sent in. The query form returns the number of minutes based on the new settings values.

### Conditions

Measurement views: Spectrogram

### Group

Trace commands

### Syntax

```
TRACe:SGRam[:MEASview<y>]:FUNctIon:TIME <value>
```

```
TRACe:SGRam[:MEASview<y>]:FUNctIon:TIME?
```

### Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<value>::<NR1> specifies the time length in Spectrum Monitor mode.

Range: 1 to 600 minutes and 60 seconds.

### Returns

<NRf> specifies the time length in Spectrum Monitor mode.

### Examples

```
TRACe:SGRam:MEASview1:FUNctIon:TIME 6E2 sets the Time/update value to 600 minutes.
```

## TRACe:SGRam[:MEASview<y>]:RESet

Determines whether or not to reset the spectrogram display.

### Conditions

Measurement view: Spectrogram

### Group

Trace commands

### Syntax

```
TRACe:SGRam[:MEASview<y>]:RESet {OFF | ON | 0 | 1}
```



## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

OFF or 0 does not reset the display of the spectrogram.

ON or 1 resets the display of the spectrogram.

## Examples

TRACe:SGRam:MEASview1:RESet ON resets the spectrogram measurement view 1 display.

## TRACe:SGRam[:MEASview<y>]:SElect:LINE

Sets or queries the line number to send to the spectrum display.

## Conditions

Measurement views: Spectrogram

## Group

Trace commands

## Syntax

TRACe:SGRam[:MEASview<y>]:SElect:LINE <number>

TRACe:SGRam[:MEASview<y>]:SElect:LINE?

## Related Commands

TRACe<x>:SPECTrum

## Arguments

<y> = 1 to 8 represents measurement index based on number of sources of connected instrument. where MEASview keyword is optional by default y is 1.

<number> ::= <NR1> specifies the line number to send to the spectrum display.

Range: 0 to the maximum line number of the spectrogram displayed on screen.

## Returns

<NRf> specifies the line number to send to the spectrum display.

## Examples

TRACe:SGRam:MEASview1:SElect:LINE 75 selects Line #75 in the spectrogram to send to the spectrum display.

## TRACe:SPURious:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for multi-trace functions (Max Hold and Average) in the Spurious measurement. This command is effective when TRACe:SPURious:FUNCTion is set to MAXHold or AVERage and INITiate:CONTinuous is set to OFF.

## Conditions

Measurement views: Spurious

## Group

Trace commands

## Syntax

```
TRACe:SPURious:COUNT <number>
```

```
TRACe:SPURious:COUNT?
```

## Arguments

<number>: : <NR1> specifies the count for multi-trace functions.

Range: 1 to 10000.

## Examples

TRACE:SPURIOUS:COUNT32 sets the count to 32 for multi-trace functions.

## TRACe:SPURious:COUNT:ENABLE

Determines whether to enable or disable the count for multi-trace functions (Max Hold and Average) in the Spurious view. This command is effective when TRACe:SPURious:FUNCTion is set to MAXHold or AVERAge.

## Conditions

Measurement views: Spurious

## Group

Trace commands

## Syntax

```
TRACe:SPURious:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe:SPURious:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count for multi-trace functions.

ON or 1 enables the count for multi-trace functions.

## Examples

TRACE:SPURIOUS:COUNT:ENABLEON enables the count for multi-trace functions.

## TRACe:SPURious:COUNT:RESet (No Query Form)

Clears the multi-function (Max Hold or Average) data and counter, and restarts the process in the Spurious view. This command is effective when TRACe:SPURious:FUNCTion is set to MAXHold or AVERAge.

## Conditions

Measurement views: Spurious

## Group

Trace commands

## Syntax

```
TRACe:SPURious:COUNT:RESet
```

## Arguments

None

## Examples

`TRACE:SPURIOUS:COUNT:RESET` clears the multi-function data and counter, and restarts the process.

## TRACe:SPURious:FREeze

Determines whether or not to freeze the display of the trace in the Spurious view.

## Conditions

Measurement views: Spurious

## Group

Trace commands

## Syntax

```
TRACe:SPURious:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe:SPURious:FREeze?
```

## Arguments

OFF or 0 updates the display of the trace normally.

ON or 1 stops updating the display of the trace.

## Examples

`TRACE:SPURIOUS:FREEZEON` stops updating the display of the trace.

## TRACe:SPURious:FUNcTion

Sets or queries the trace function in the Spurious view.

## Conditions

Measurement views: Spurious

## Group

Trace commands

## Syntax

```
TRACe:SPURious:FUNCTion { NONE | MAXHold | AVERage | AVGLog}
```

```
TRACe:SPURious:FUNCTion?
```

## Arguments

NONE selects normal display.

MAXHold selects the Max Hold display that indicates the maximum amplitude drift at each frequency point.

AVERage selects the Average display that indicates the average amplitude drift at each frequency point.

AVGLog selects the Average of logs display that indicates the average amplitude drift of logs at each frequency point.

## Examples

TRACE:SPURIOUS:FUNCTIONMAXHold displays the Max Hold trace in the Spurious measurement.

## TRACe:WLAN:AD:CONStE:TYPE

Sets or queries the trace type in the 802.11ad Constellation display.

## Conditions

Measurement view: 802.11ad Constellation

## Group

Trace commands

## Syntax

```
TRACe:WLAN:AD:CONStE:TYPE { ROTA | REGU }
```

```
TRACe:WLAN:AD:CONStE:TYPE?
```

## Arguments

ROTA is  $\pi/2$  Rotated trace type

REGU is the Regular trace type

## Returns

ROTA means the trace type is set to  $\pi/2$  rotated REGU means the trace type is set to regular

## Examples

TRACE:WLAN:AD:CONStE:TYPE ROTA sets the trace type to  $\pi/2$  Rotated.

## TRACe:WLAN:AY:CONStE:TYPE

Set or query the Trace Type (Regular or PI/2 Rotated).

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Trace commands

### Syntax

```
TRACe:WLAN:AY:CONStE:TYPE { REGU | ROTA }
```

```
TRACe:WLAN:AY:CONStE:TYPE?
```

### Arguments

REGU sets the Trace Type as Regular.

ROTA sets the Trace Type as PI/2 Rotated.

### Returns

REGU means that the Trace Type is Regular.

ROTA means that the Trace Type is PI/2 Rotated

### Examples

TRAC:WLAN:AY:CONS:TYPE REGU sets the Trace Type to Regular.

TRAC:WLAN:AY:CONS:TYPE? may return ROTA, which means the Trace Type is PI/2 Rotated.

## TRACe:WLAN:FLATness:FREeze

Sets or queries whether to freeze the average trace on the WLAN Spectral Flatness display. When the freeze state is turned on, the selected trace stops updating.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Trace commands

### Syntax

```
TRACe:WLAN:FLATness:FREeze { 0 | 1 | OFF | ON }
```

```
TRACe:WLAN:FLATness:FREeze?
```

## Arguments

ON or 1 turns on the freeze state, causing the selected trace to stop updating.

OFF or 0 turns off the freeze state.

## Returns

Outputs the frozen status of the selected trace as an integral value.

1 indicates a frozen trace and 0 a non-frozen trace.

## Examples

TRAC : WLAN : FLAT : FRE 1 freezes the trace, causing it to stop updating.

## TRACe:WLAN:FLATness:SHOW

Displays or hides the trace on the WLAN Spectral Flatness display.



**Note:** WLAN Spectral Flatness is only available for OFDM (non-802.11b) signals.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe:WLAN:FLATness:SHOW {0|1|OFF|ON}
```

```
TRACe:WLAN:FLATness:SHOW?
```

## Arguments

ON or 1 displays the trace.

OFF or 0 hides the trace.

## Examples

TRACe:WLAN:FLAT:SHOW 1 turns on the display of the trace.

## TRACe1:TOverview[:MEASview<y>]

Enables display of or queries the display status of the specified trace. Only Trace1 is valid.

## Conditions

Measurement views: General Waveform display

## Group

Trace commands

## Syntax

```
TRACe1:TOVerview:MEASview<y> { OFF | ON | 0 | 1 }
```

```
TRACe1:TOVerview:MEASview<y>?
```

## Arguments

OFF or 0 disables the trace overview.

ON or 1 enables the trace overview.

## Examples

TRACe1:TOVerview:MEASVIEW1 shows the trace1 in the Time overview Measurement View 1.

## TRACe1:TOVerview[:MEASview<y>]:AVERage:COUNT

Sets or queries the number of traces averaged to generate the specified trace. Only Trace1 is valid.

## Conditions

Measurement views: Trace Function set to Average

## Group

Trace commands

## Syntax

```
TRACe1:TOVerview[:MEASview<y>]:AVERage:COUNT <value>
```

```
TRACe1:TOVerview[:MEASview<y>]:AVERage:COUNT?
```

## Arguments

<value>::<NR1> an integer number of traces to average to create the waveform display.

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<NRf>

## Examples

TRACe1:TOVerview:MEASview1:AVERage:COUNT 200 sets the Average count for Trace 1 to 200.

## TRACe1:TOVerview[:MEASview<y>]:COUNT

Enables or queries the count set for the specified trace. Only Trace1 is valid.

## Conditions

Measurement views: Trace Function set to Average

## Group

Trace commands

## Syntax

```
TRACe1:TOVerview[:MEASview<y>]:COUNT <value>
```

```
TRACe1:TOVerview[:MEASview<y>]:COUNT?
```

## Arguments

<value>::<Nrf>

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

## Returns

<Nrf>

## Examples

TRACe1:TOVerview:MEASview1:COUNT 200 sets the Trace1 Average count to 200.

## TRACe1:TOVerview[:MEASview<y>]:COUNT:ENABLE

Enables or queries the Average count for the specified trace. Only Trace1 is valid.

## Conditions

Measurement views: Trace Function set to Average

## Group

Trace commands

## Syntax

```
TRACe1:TOVerview[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe1:TOVerview[:MEASview<y>]:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the trace count.

ON or 1 enables the trace count.

## Examples

TRACe1:TOVerview:MEASview1:COUNT:ENABLE ON enables the Trace1 Average count.



## TRACe1:TOVerview[:MEASview<y>]:COUNT:RESet (No Query Form)

Resets the waveform count for the specified trace. Only Trace1 is valid.

### Conditions

Measurement views: Trace Function set to Average

### Group

Trace commands

### Syntax

```
TRACe1:TOVerview[:MEASview<y>]:COUNT:RESet
```

### Arguments

The parameter <y> = 1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value is 1.

### Examples

TRACe1:TOVerview:MEASview1:COUNT:RESet sets the trace count to 1.

## TRACe1:TOVerview[:MEASview<y>]:DETection

Enables or queries the type of detection for the specified trace. Only Trace1 is valid.

### Conditions

Measurement views: all

### Group

Trace commands

### Syntax

```
TRACe1:TOVerview[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative | POSNegative | SAMPlE }
```

```
TRACe1:TOVerview[:MEASview<y>]:DETection?
```

### Arguments

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

## Examples

TRACe1:TOVerview:MEASview1:DETection POSitive enables positive detection on Trace1.

## TRACe1:TOVerview[:MEASview<y>]:FREeze

Enables or queries a halt to acquisition updates for the specified trace. Only Trace1 is valid.

### Conditions

Measurement views: all

### Group

Trace commands

### Syntax

```
TRACe1:TOVerview[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
TRACe1:TOVerview[:MEASview<y>]:FREeze?
```

### Arguments

OFF or 0 disables the trace freeze function.

ON or 1 enables the trace freeze function.

## Examples

TRACe1:TOVerview:MEASview1:FREeze ON Halts acquisition updates to Trace1.

## TRACe1:TOVerview[:MEASview<y>]:FUNCTION

Enables or queries the selected Function for the specified trace. Only Trace1 is valid.

### Conditions

Measurement views: all

### Group

Trace commands

### Syntax

```
TRACe1:TOVerview[:MEASview<y>]:FUNCTION { NORMAl | AVERAge | MAXHold | MINHold }
TRACe1:TOVerview[:MEASview<y>]:FUNCTION?
```

### Arguments

NORMAl Each new trace is displayed and then replaced by the next trace.

AVERAge Multiple traces are averaged together to generate the displayed trace. MAXHold Displays the maximum value in the trace record for each display point. MINHold Displays the minimum value in the trace record for each display point.

## Examples

TRACe1:TOVerview:MEASview1:FUNCTION AVERAge enables the averaging of multiple traces on Trace1.

## TRACe<x>:{AM|FM|PM}

Determines whether or not to show the trace in the AM/FM/PM view.

### Conditions

Measurement views: AM, FM, PM

### Group

Trace commands

### Syntax

```
TRACe<x>:{AM|FM|PM} { OFF | ON | 0 | 1 }
```

```
TRACe<x>:{AM|FM|PM}?
```

### Arguments

OFF or 0 hides the trace in the AM, FM, or PM view.

ON or 1 shows the trace in the AM, FM, or PM view.

## Examples

TRACE:AM ON shows the trace in the AM view.

## TRACe<x>:AVTime[:MEASview<y>]

Determines whether or not to show the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 4; All traces are valid.

### Conditions

Measurement views: Amplitude versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:AVTime[:MEASview<y>] { OFF | ON | 0 | 1 }
```

```
TRACe<x>:AVTime[:MEASview<y>]?
```

### Arguments

OFF or 0 hides the specified trace in the Amplitude versus Time view.

ON or 1 shows the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1 ON shows Trace 1 in the Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:AVERage:COUNT

Sets or queries the number of traces to combine. This command is effective when you select AVERage with the TRACe<x>:AVTime[:MEASview<y>]:FUNction command.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

TRACe<x>:AVTime[:MEASview<y>]:AVERage:COUNT <number>

TRACe<x>:AVTime[:MEASview<y>]:AVERage:COUNT?

## Arguments

<number>: : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:AVERAGE:COUNT 64 sets the average count to 64 for Trace 1 for Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:AVERage:RESet (No Query Form)

Restarts acquisition and display of waveforms for the specified trace. For an Average, Max Hold, or Min Hold trace, it restarts the sequence, discarding accumulated data and resetting the counter.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:AVERage:RESet
```

## Related Commands

```
TRACe<x>:AVTime[:MEASview<y>]:FUNctIon
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:AVERAGE:RESET restarts acquisition and display of waveforms for Trace 1 on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Amplitude versus Time measurement. This command is effective when TRACe<x>:AVTime[:MEASview<y>]:FUNctIon is set to MAXHold or MINHold and INITiate:CONTInuous is set to OFF.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:COUNT <number>
```

```
TRACe<x>:AVTime[:MEASview<y>]:COUNT?
```

## Arguments

<number>::<NR1> specifies the count for Max/Min Hold.

Range: 1 to 10000.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:COUNT 32 sets the count to 32 for Trace 1 on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:COUNT:ENABLE

Determines whether to enable or disable the count for the Max or Min Hold trace in the Amplitude versus Time view. This command is effective when TRACe<x>:AVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:AVTime[:MEASview<y>]:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count for the Max/Min Hold trace.

ON or 1 enables the count for the Max/Min Hold trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:COUNT:ENABLE ON enables the Max/Min Hold count for Trace 1 on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:COUNT:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process for the specified trace in the Amplitude versus Time view. This command is effective when TRACe<x>:AVTime[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:COUNT:RESet
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:COUNT:RESET clears the Max/Min Hold data and counter, and restarts the process for Trace 1 on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:DETection

Sets or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:DETection { AVERage | POSitive | NEGative | POSNegative | SAMPlE }
```

```
TRACe<x>:AVTime[:MEASview<y>]:DETection?
```

## Arguments

AVERage displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE:AVTIME:MEASVIEW1:DETECTION AVERage specifies that the trace displays the average data value for each pixel on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:FREeze

Determines whether or not to freeze the display of the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 4; All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:AVTime[:MEASview<y>]:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:AVTIME:MEASVIEW1:FREeze ON freezes the display for Trace 1 in the Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:FUNCTION

Sets or queries the function for the specified trace in the Amplitude versus Time view.

The parameter <x> = 1 to 3; Trace 4 (math trace) is invalid.

## Conditions

Measurement views: Amplitude versus Time



## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:FUNctIon { NORMAl | AVERAge | MAXHold | MINHold }
TRACe<x>:AVTime[:MEASview<y>]:FUNctIon?
```

## Arguments

**NORMAl** selects the normal display.

**AVERAge** selects the Average display that indicates the average amplitude at each time point.

**MAXHold** selects the Max Hold display that indicates the maximum amplitude at each time point.

**MINHold** selects the Min Hold display that indicates the minimum amplitude at each time point.

The parameter <x> = 1 to 4; **MARKer0** (reference marker) is invalid. The specified marker must be activated using the **CALCulate:MARKer:ADD** command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where **MEASview** keyword is optional by default y is 1

## Examples

**TRACE1:AVTIME:MEASVIEW1:FUNCTION MAXHold** selects Max Hold for Trace 1 in the Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:LEFToperand

Sets or queries the left operand for the math trace (Trace 4) in the Amplitude versus Time view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:LEFToperand { TRACE1 | TRACE2 | TRACE3 }
TRACe<x>:AVTime[:MEASview<y>]:LEFToperand?
```

## Related Commands

**TRACe<x>:AVTime[:MEASview<y>]:RIGHToperand**

## Arguments

**TRACE1** selects Trace 1 as the left operand for the math trace. **TRACE2** selects Trace 2 as the left operand for the math trace. **TRACE3** selects Trace 3 as the left operand for the math trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE2 : AVTIME : MEASVIEW1 : LEFTOPERAND selects Trace 2 as the left operand for the math trace for Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:RIGHtoperand

Sets or queries the right operand for the math trace (Trace 4) in the Amplitude versus Time view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:RIGHtoperand { TRACE1 | TRACE2 | TRACE3 }
```

```
TRACe<x>:AVTime[:MEASview<y>]:RIGHtoperand?
```

## Related Commands

```
TRACe<x>:AVTime[:MEASview<y>]:LEFTooperand
```

## Arguments

TRACE1 selects Trace 1 as the right operand for the math trace. TRACE2 selects Trace 2 as the right operand for the math trace. TRACE3 selects Trace 3 as the right operand for the math trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1 : AVTIME : MEASVIEW1 : RIGHTOPERAND selects Trace 1 as the right operand for the math trace on Amplitude versus Time Measurement View 1.

## TRACe<x>:AVTime[:MEASview<y>]:SELEct

Sets or queries the trace whose measurement results are being displayed in the readout on the top and bottom of the view.

The parameter <x> = 1 to 4; All traces are valid.

## Conditions

Measurement views: Amplitude versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:AVTime[:MEASview<y>]:SElect
```

```
TRACe<x>:AVTime[:MEASview<y>]:SElect?
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Returns

```
{ 0 | 1 }
```

- 0 indicates that the results are not being displayed in the readout for the trace.
- 1 indicates that the results are being displayed in the readout for the trace.

## Examples

TRACE1:AVTIME:MEASVIEW1:SELECT selects Trace 1 to display the measurement results in the readout for Amplitude versus Time Measurement View 1.

TRACE1:AVTIME:MEASVIEW1:SELECT? might return 1, indicating that the results are being displayed in the readout for Trace 1 on Amplitude versus Time Measurement View 1.

## TRACe<x>:BLUetooth:CONSte

Sets or queries the specified trace to display or hide in the Bluetooth Constellation display.

The parameter <x> is trace 1 or 2.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Trace commands

## Syntax

```
TRACe<x>:BLUetooth:CONSte { OFF | ON | 0 | 1 }
```

```
TRACe<x>:BLUetooth:CONSte?
```

## Arguments

OFF or 0 hides the selected trace.

ON or 1 displays the selected trace.

## Returns

0 indicates the specified trace is not shown in the Bluetooth Constellation display.

1 indicates the specified trace is shown in the Bluetooth Constellation display.

## Examples

`TRACe1:BLUEtooth:CONSte ON` specifies that trace 1 appears in the Bluetooth Constellation display.

## TRACe<x>:BLUEtooth:CONSte:FREeze

Sets or queries the specified trace to freeze or not to freeze in the Bluetooth Constellation display.

The parameter <x> is trace 1 or 2.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Trace commands

## Syntax

```
TRACe<x>:BLUEtooth:CONSte:FREeze {OFF | ON | 0 | 1 }
```

```
TRACe<x>:BLUEtooth:CONSte:FREeze?
```

## Arguments

OFF or 0 updates the display of the selected trace normally.

ON or 1 stops updating the display of the selected trace.

## Returns

0 indicates the specified trace is being updated in the Bluetooth Constellation display.

1 indicates the specified trace is not being updated in the Bluetooth Constellation display.

## Examples

`TRACe1:BLUEtooth:CONSte:FREeze ON` freezes the display for trace 1 in the Bluetooth Constellation display.

## TRACe<x>:BLUEtooth:CONSte:MODE

Sets or queries the Bluetooth Constellation trace display mode. The parameter <x> is trace 1 or 2.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Trace commands

## Syntax

```
TRACe<x>:BLUEtooth:CONStE:MODE { VECT | SYMB | LIN }
```

```
TRACe<x>:BLUEtooth:CONStE:MODE?
```

## Arguments

VECT connects adjacent symbol points with the signal locus.

SYMB displays individual symbol points. LIN displays linear symbol points.

## Examples

TRACe1:BLUEtooth:CONStE:MODE VECT shows the Bluetooth Constellation display is connecting adjacent symbol points with the signal locus.

## TRACe<x>:BLUEtooth:CONStE:SELEct

Sets or queries the active trace in the Bluetooth Constellation diagram. The parameter <x> is trace 1 or 2.

## Conditions

Measurement views: Bluetooth Constellation

## Group

Trace commands

## Syntax

```
TRACe<x>:BLUEtooth:CONStE:SELEct <active>
```

```
TRACe<x>:BLUEtooth:CONStE:SELEct?
```

## Arguments

<active> ::= {0 | 1}

Where:

0, the specified trace is not the selected trace.

1, the specified trace is the selected trace.

## Examples

TRACe1:BLUEtooth:CONStE:SELEct specifies trace 1 as the selected trace.

## TRACe<x>:BLUetooth:EDIagram:ENABLE:I

Determines whether to show or hide the I trace in the Bluetooth eye diagram.

### Conditions

Measurement views: Bluetooth Eye diagram

### Group

Trace commands

### Syntax

```
TRACe<x>:BLUetooth:EDIagram:ENABLE:I { OFF | ON | 0 | 1 }
```

```
TRACe<x>:BLUetooth:EDIagram:ENABLE:I?
```

### Arguments

OFF or 0 hides the I trace.

ON or 1 shows the I trace.

### Examples

```
TRACEX:BLUETOOTH:EDIAGRAM:ENABLE:I ON shows the I trace in the eye diagram.
```

## TRACe<x>:BLUetooth:EDIagram:ENABLE:Q

Determines whether to show or hide the Q trace in the Bluetooth Eye diagram.

### Conditions

Measurement views: Bluetooth Eye diagram

### Group

Trace commands

### Syntax

```
TRACe<x>:BLUetooth:EDIagram:ENABLE:Q { OFF | ON | 0 | 1 }
```

```
TRACe<x>:BLUetooth:EDIagram:ENABLE:Q?
```

### Arguments

OFF or 0 hides the Q trace.

ON or 1 shows the Q trace.

### Examples

```
TRACEX:BLUETOOTH:EDIAGRAM:ENABLE:Q ON shows the Q trace in the eye diagram.
```

## TRACe<x>:BLUEtooth:EDIagram:SElect:I

Selects the I trace in the Bluetooth eye diagram. The query version of this command returns whether the I trace is selected or not.

### Conditions

Measurement views: Bluetooth Eye diagram

### Group

Trace commands

### Syntax

```
TRACe<x>:BLUEtooth:EDIagram:SElect:I { 0 | 1 }
```

```
TRACe<x>:BLUEtooth:EDIagram:SElect:I?
```

### Arguments

None

### Returns

0 indicates that the I trace is deselected.

1 indicates that the I trace is selected.

### Examples

TRACEX:BLUETOOTH:EDIAGRAM:SELECT:I selects the I trace in the eye diagram.

## TRACe<x>:BLUEtooth:EDIagram:SElect:Q

Selects the Q trace in the Bluetooth eye diagram. The query version of this command returns whether the I trace is selected or not.

### Conditions

Measurement views: Bluetooth Eye diagram

### Group

Trace commands

### Syntax

```
TRACe<x>:BLUEtooth:EDIagram:SElect:Q { 0 | 1 }
```

```
TRACe<x>:BLUEtooth:EDIagram:SElect:Q?
```

### Arguments

None

### Returns

0 indicates that the Q trace is deselected.

1 indicates that the Q trace is selected.

## Examples

`TRACEx:BLUETOOTH:EDIAGRAM:SELECT:Q` selects the Q trace in the eye diagram.

## TRACe<x>:BLUEtooth:FDVTime:MODE

Sets or queries whether to display the Frequency Deviation vs. Time trace as vectors or symbols (points).

### Conditions

Measurement views: Frequency Deviation vs. Time

### Group

Trace commands

### Syntax

```
TRACe<x>:BLUEtooth:FDVTime:MODE { VECTors | SYMBols }
```

```
TRACe<x>:BLUEtooth:FDVTime:MODE?
```

### Arguments

`VECTors` connects adjacent symbol points with lines.

`SYMBols` displays individual symbol points.

## Examples

`TRACe:BLUEtooth:FDVTime:MODE VECT` sets the display to connect points in the display with lines.

## TRACe<x>:CCDF:FREeze

Determines whether or not to freeze the display of the specified trace (Trace 1 or 2) in the CCDF view.

The parameter <x> = 1 or 2; Trace 3 (Gaussian curve) is invalid.

### Conditions

Measurement views: CCDF

### Group

Trace commands

### Syntax

```
TRACe<x>:CCDF:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:CCDF:FREeze?
```

### Arguments

`OFF` or `0` updates the display of the specified trace normally.



ON or 1 stops updating the display of the specified trace.

## Examples

TRACE1 : CCDF : FREEZE ON freezes the display for Trace 1.

## TRACe<x>:CCDF:SElect

Sets or queries the trace whose measurement results are being displayed in the readout on the top of the view. The selected trace is indicated by the measurement pointer (pink triangle) on the waveform.

The parameter <x> = 1 to 3; All traces are valid.

## Conditions

Measurement views: CCDF

## Group

Trace commands

## Syntax

TRACe<x>:CCDF:SElect

TRACe<x>:CCDF:SElect?

## Arguments

None

## Returns

{ 0 | 1 }

0 indicates that the results are not being displayed in the readout for the trace.

1 indicates that the results are being displayed in the readout for the trace.

## Examples

TRACE1 : CCDF : SELECT selects Trace 1 to display the measurement results in the readout.

TRACE1 : CCDF : SELECT? might return 1, indicating that the results are being displayed in the readout for Trace 1.

## TRACe<x>:CCDF:SHOW

Determines whether to show or hide the specified trace in the CCDF view. The parameter <x> = 1 to 3; All traces are valid.

## Conditions

Measurement views: CCDF

## Group

Trace commands

## Syntax

```
TRACe<x>:CCDF:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:CCDF:SHOW?
```

## Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

## Examples

TRACE1:CCDF:SHOW ON shows Trace 1 in the CCDF view.

## TRACe<x>:CCDF:X

Sets or queries the horizontal position of the measurement pointer (pink triangle) to measure the CCDF. Use the TRACe<x>:CCDF:Y? query to read the value.

The parameter <x> = 1 to 3.



**Note:** Use the TRACe<x>:CCDF:SHOW command to show the specified trace and the TRACe<x>:CCDF:SElect command to select the trace before running the TRACe<x>:CCDF:X command.

## Conditions

Measurement views: CCDF

## Group

Trace commands

## Syntax

```
TRACe<x>:CCDF:X <value>
```

```
TRACe<x>:CCDF:X?
```

## Arguments

<value>::<Nrf> specifies the horizontal position of the measurement pointer.

Range: 0 to 20 dB.

## Examples

TRACE1:CCDF:X 5 puts the measurement pointer at 5 dB on Trace 1.

## TRACe<x>:CCDF:Y? (Query Only)

Queries the vertical position (CCDF value) of the measurement pointer (displayed as a pink triangle). Use the TRACe<x>:CCDF:X command to set the horizontal position of the pointer.

The parameter <x> = 1 to 3.



**Note:** Use the `TRACe<x>:CCDF:SHOW` command to show the specified trace and the `TRACe<x>:CCDF:SElect` command to select the trace before running the `TRACe<x>:CCDF:Y?` query.

## Conditions

Measurement views: CCDF

## Group

Trace commands

## Syntax

`TRACe<x>:CCDF:Y?`

## Arguments

None

## Returns

`<value>: : <NRf>` is the vertical position (CCDF) of the measurement pointer.

Range: 0 to 100%.

The value of 99.0999953003E+36 is returned if the trace is not available.

## Examples

`TRACE1:CCDF:Y?` might return 14.72, indicating the CCDF is 14.72% at the measurement pointer on Trace 1.

## TRACe<x>:CONSte[:MEASview<y>]

Specifies whether or not to show the specified trace (Trace 1 or 2) in the Constellation display.

The parameter `<x>` = 1 or 2.

The parameter `<y>` = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Trace commands

## Syntax

`TRACe<x>:CONSte[:MEASview<y>] {OFF | ON | 0 | 1 }`

`TRACe<x>:CONSte[:MEASview<y>]?`

## Arguments

OFF or 0 hides the selected trace.

ON or 1 displays the selected trace.

## Returns

{ 0 } indicates the specified trace is not shown in the Constellation display.

{ 1 } indicates the specified trace is shown in the Constellation display.

## Examples

TRACe1:CONSte:MEASview2 2 specifies that Trace 2 be shown in the Constellation display for measview 2 of constellation.

## TRACe<x>:CONSte[:MEASview<y>]:FREeze

Specifies whether or not to freeze (halt updates) to the display of the specified trace (Trace 1 or 2) in the Constellation display.

The parameter <x> = 1 or 2.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation) .

## Group

Trace commands

## Syntax

```
TRACe<x>:CONSte[:MEASview<y>]:FREeze {OFF | ON | 0 | 1 }
```

```
TRACe<x>:CONSte[:MEASview<y>]:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

## Returns

{ 0 } indicates the specified trace is being updated.

{ 1 } indicates the specified trace is not being updated.

## Examples

TRACe1:CONSte:MEASview2:FREeze ON freezes the display for Trace 1 for measview 2 of constellation.

## TRACe<x>:CONSte[:MEASview<y>]:MODE

Sets or queries how to display the constellation trace.

The parameter <x> = 1 or 2.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

### Group

Trace commands

### Syntax

```
TRACe<x>:CONSte[:MEASview<y>]:MODE { VECTors | SYMBols }
TRACe<x>:CONSte[:MEASview<y>]:MODE?
```

### Arguments

VECTors connects adjacent symbol points with the signal locus.

SYMBols displays individual symbol points.

### Examples

TRACe<X>:CONSte:MEASview2:MODE VECTors shows the constellation connecting adjacent symbol points with the signal locus for measview 2 of constellation.

## TRACe<x>:CONSte[:MEASview<y>]:Q:OFFSet

Sets the Q offset in the Constellation display to Use shared Pref, Remove Q offset, or Include Q offset.

The parameter <x> = 1 or 2.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

### Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

### Group

Trace commands

## Syntax

```
TRACe<x>:CONStE[:MEASview<y>]:Q:OFFSet { SHARed | RQOFset | IQOFset }
TRACe<x>:CONStE[:MEASview<y>]:Q:OFFSet?
```

## Arguments

SHARed means that the Q offset applied to the selected trace will be the same as the Remove Q offset setting located on the Prefs tab (which applies to all GP Digital Modulation displays).

Remove Q offset removes the Q offset, possibly overriding the Remove Q offset setting on the Prefs tab. Changing the Constellation display Q Offset value with this command does not affect the Q offset setting on other displays.

Include Q offset includes the Q offset, possibly overriding the Remove Q offset setting on the Prefs tab. Changing the Constellation display Q Offset value with this command does not affect the Q offset setting on other displays.

## Returns

{ SHAR } indicates that Q Offset is set to Use Shared Pref.

{ RQOF } indicates that Q Offset is set to Remove Q Offset.

{ IQOF } indicates that Q Offset is set to Include Q Offset.

## Examples

TRACe1:CONStE:MEASview2:Q:OFFSet SHAR sets the Constellation display Q Offset to Use Shared Pref for measview 2 of constellation.

## TRACe<x>:CONStE[:MEASview<y>]:SELEct (No query form)

Sets or queries the trace to which the Show, Freeze, Content, and Q Offset settings apply.

The parameter <x> = 1 or 2.

The parameter <y> = 1 to number of instances of a measurement created or number of channels available in an instrument (Example: 1 to 8 channels for MSO58).

## Conditions

Measurement views: Constellation

MEASview keyword is optional and applicable only for measurements that supports multi-instance creation of measurement (Example: GPDemod Constellation).

## Group

Trace commands

## Syntax

```
TRACe<x>:CONStE[:MEASview<y>]:SELEct
TRACe<x>:CONStE[:MEASview<y>]:SELEct?
```

## Arguments

None

## Returns

{ 0 } indicates the specified trace is not the selected trace.

{ 1 } indicates the specified trace is the selected trace.

## Examples

`TRACe1:CONStE:MEASview2:SELEct` specifies Trace 1 as the selected trace for measview 2 of constellation.

## TRACe<x>:DPX

Determines whether or not to show the specified trace in the DPX, Spectrum view.

The parameter <x> = 1 to 7; All traces are valid. Trace 6 is for the DPXogram trace and Trace 7 is for the Ogram line.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

`TRACe<x>:DPX { OFF | ON | 0 | 1 }`

`TRACe<x>:DPX?`

## Arguments

OFF or 0 hides the specified trace in the DPX, Spectrum view.

ON or 1 shows the specified trace in the DPX, Spectrum view.

## Examples

`TRACe1:DPX ON` shows Trace 1 (the maximum trace) in the DPX, Spectrum view.

## TRACe<x>:DPX:AVERAge:COUNT

Sets or queries the number of traces to combine for averaging in the DPX, Spectrum view.

The parameter <x> = 3; Only Trace 3 (average trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

`TRACe<x>:DPX:AVERAge:COUNT <number>`

TRACe<x>:DPX:AVERAge:COUNT?

### Arguments

<number>: : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

### Examples

TRACE3:DPX:AVERAGE:COUNT 32 sets the average count to 32.

## TRACe<x>:DPX:COLor:CURVe

Sets or queries how colors are mapped to the signal density in the DPX, Spectrum bitmap display. The mapping can be linear (Curve = 1), or it can be set to concentrate the resolution on the lower level of the range (Curve > 1) or the mapping can be set to show the best resolution on the upper range of density or hit count (Curve = 0.1 – 0.99). This command is only valid for the Bitmap display.

The parameter <x> = 5; only Trace5 is valid.

### Conditions

Measurement views: DPX, Spectrum

### Group

Trace commands

### Syntax

TRACe<x>:DPX:COLor:CURVe <value>

TRACe<x>:DPX:COLor:CURVe?

### Arguments

<value>: : <NRf> specifies how colors are mapped to the signal density. Range:

### Examples

TRACe<x>:DPX:COLor:CURVe 1.5 concentrates the resolution on the lower level of the range on the Bitmap display.

## TRACe<x>:DPX:COLor:INTensity

Sets or queries the color intensity in the DPX, Spectrum view. The value is common to all traces.

The parameter <x> = 1 to 5; All traces are valid.

### Conditions

Measurement views: DPX, Spectrum

### Group

Trace commands



## Syntax

```
TRACe<x>:DPX:COLor:INTensity <value>
```

```
TRACe<x>:DPX:COLor:INTensity?
```

## Arguments

<value>::<Nrf> specifies color intensity. Range: 1 to 100%.

## Examples

TRACE1:DPX:COLOR:INTENSITY 30 sets the color intensity to 30%.

## TRACe<x>:DPX:COLor:SCALE:AUTO (No Query Form)

Automatically adjusts the Max and Min color settings to display the broadest range of colors in the DPX, Spectrum bitmap display. This command is only valid for the Bitmap display.

The parameter <x> = 5; only Trace5 is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:COLor:SCALE:AUTO
```

## Arguments

<x>::=5 specifies the auto-color setting for the Bitmap trace.

## Examples

TRACE5:DPX:COLOR:SCALE:AUTO autosets the Bitmap display color scheme.

## TRACe<x>:DPX:DETection

Sets or queries the detection type (POSitive, NEGative, AVERage) for traces 1, 2 and 3 in the DPX, Spectrum view.

Trace 4 (math trace) and Trace 5 (bitmap trace) are invalid choices. Trace 6 is only valid for DPXogram.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:DETection { AVERAge | NEGative | POSitive }
```

```
TRACe<x>:DPX:DETection?
```

## Arguments

The following table shows the trace function and display. For the average trace, use the `TRACe<x>:DPX:AVERAge:COUNT` command to set the average count.

## Examples

```
TRACE1:DPX:DETection AVERAge enables analysis to look for AVERAge on Trace 1.
```

## TRACe<x>:DPX:DGRAM:SElect:LINE

Sets or queries a line from the DPXogram display to send to the DPX, Spectrum display.

Trace 6 is the DPXogram trace. No other traces are allowed.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:DGRAM:SElect:LINE <number>
```

```
TRACe<x>:DPX:DGRAM:SElect:LINE?
```

## Arguments

`<number>` ::= `<NR1>` specifies the line number in the DPXogram display.

## Examples

```
TRACE6:DPX:DGRAM:SELECT:LINE 75 selects line #75 in the DPXogram display to send to the DPX, Spectrum display.
```

## TRACe<x>:DPX:DOT:PERSistent

Determines whether to enable or disable the dot persistence for the bitmap trace (Trace 5) in the DPX, Spectrum view.

The parameter `<x>` = 5; Only Trace 5 (bitmap trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:DOT:PERSiStent { OFF | ON | 0 | 1 }
```

```
TRACe<x>:DPX:DOT:PERSiStent?
```

## Arguments

OFF or 0 disables the dot persistence.

ON or 1 enables the dot persistence.

## Examples

TRACE5:DPX:DOT:PERSISTENT ON enables the dot persistence in the DPX, Spectrum view.

## TRACe<x>:DPX:DOT:PERSiStent:TYPE

Sets or queries the persistence type for the bitmap trace (Trace 5) in the DPX, Spectrum view.

The parameter <x> = 5; Only Trace 5 (bitmap trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:DOT:PERSiStent:TYPE { VARIable | INFInite }
```

```
TRACe<x>:DPX:DOT:PERSiStent:TYPE?
```

## Arguments

VARIable selects the variable persistence display which leaves acquired data points on the display for a period of time specified by the TRACe<x>:DPX:DOT:PERSiStent:VARIable command.

INFInite selects the infinite persistence display which accumulates data points on the display indefinitely.

## Examples

TRACE5:DPX:DOT:PERSISTENT:TYPE VARIable selects the variable persistence display.

## TRACe<x>:DPX:DOT:PERSiStent:VARIable

Sets or queries how long data points are displayed. This command is effective when TRACe<x>:DPX:DOT:PERSiStent:TYPE is set to VARIable. This affects the display only.

The parameter <x> = 5; Only Trace 5 (bitmap trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:DOT:PERSistent:VARiable <number>
```

```
TRACe<x>:DPX:DOT:PERSistent:VARiable?
```

## Arguments

<number>: : <NR1> specifies the how long waveform points are displayed on the screen. Range: 100 ms to 60 s (the default value is 1.0 s).

## Examples

TRACE5:DPX:DOT:PERSISTENT:VARIABLE 20 specifies that the waveform points are displayed on the screen for a period of 20 before they disappear.

## TRACe<x>:DPX:FREeze

Determines whether or not to freeze the display of the specified trace in the DPX, Spectrum view.

The parameter <x> = 1 to 6; All traces are valid. Trace 6 is for the DPXogram trace and Trace 7 is for the Ogram line.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:DPX:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

## Examples

TRACE1:DPX:FREEZE ON freezes the display for the +peak trace.

## TRACe<x>:DPX:FUNCTion

Sets or queries the trace function for the +Peak, -Peak, or Average trace (Trace 1, 2, or 3, respectively) in the DPX, Spectrum view.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (bitmap trace) are invalid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:FUNCTION { NORMAl | HOLD | AVERAge }
```

```
TRACe<x>:DPX:FUNCTION?
```

## Arguments

The following table shows the trace function and display. For the average trace, use the `TRACe<x>:DPX:AVERAge:COUNT` command to set the average count.

Function	Trace1 (+Peak trace)	Trace1 (+Peak trace)	Trace3 (Average trace)
NORMAl	Normal spectrum (Detection: +Peak)	Normal spectrum (Detection: - Peak)	Normal spectrum (Detection: Average)
HOLD	Max-hold spectrum	Min-hold spectrum	NA
AVERAge	NA	NA	Average spectrum

## Examples

`TRACE1:DPX:FUNCTION HOLD` selects the max hold waveform for Trace 1.

## TRACe<x>:DPX:LEFToperand

Sets or queries the left operand for the math trace (Trace 4) in the DPX, Spectrum view.

The parameter `<x>` = 4; Only Trace 4 (math trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:LEFToperand { TRACe1 | TRACe2 | TRACe3 }
```

```
TRACe<x>:DPX:LEFToperand?
```

## Related Commands

```
TRACe<x>:DPX:RIGHToperand
```

## Arguments

TRACE1 selects Trace 1 as the left operand for the math trace. TRACE2 selects Trace 2 as the left operand for the math trace. TRACE3 selects Trace 3 as the left operand for the math trace.

## Examples

TRACE4:DPX:LEFTOPERAND TRACE2 selects Trace 2 as the left operand for the math trace.

## TRACe<x>:DPX:RIGHToperand

Sets or queries the right operand for the math trace (Trace 4) in the DPX, Spectrum view.

The parameter <x> = 4; Only Trace 4 (math trace) is valid.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:RIGHToperand { TRACE1 | TRACE2 | TRACE3 }
```

```
TRACe<x>:DPX:RIGHToperand?
```

## Related Commands

```
TRACe<x>:DPX:LEFToperand
```

## Arguments

TRACE1 selects Trace 1 as the right operand for the math trace. TRACE2 selects Trace 2 as the right operand for the math trace. TRACE3 selects Trace 3 as the right operand for the math trace.

## Examples

TRACE4:DPX:RIGHTOPERAND TRACE1 selects Trace 1 as the right operand for the math trace.

## TRACe<x>:DPX:SElect

Sets or queries the trace to display the readout at the upper left of the DPX, Spectrum view.

The parameter <x> = 1 to 7; All traces are valid. Trace 6 is for the DPXogram trace and Trace 7 is for the Ogram line.

## Conditions

Measurement views: DPX, Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:DPX:SElect
```

```
TRACe<x>:DPX:SElect?
```

## Arguments

None

## Returns

```
{ 0 | 1 }
```

- 0 indicates that the readout is not being displayed for the specified trace.
- 1 indicates that the readout is being displayed for the specified trace.

## Examples

TRACE1:DPX:SELECT selects Trace 1 (+Peak trace) to display the readout.

TRACE1:DPX:SELECT? might return 1, indicating that the readout is being displayed for Trace 1.

## TRACe<x>:EMC:RECALL:ENABLE

Sets or queries that selected recalled trace is enabled (displayed) or disabled in EMC View.

<x> = <NR1> either 1,2,3,4,5 (trace number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe<x>:EMC:RECALL:ENABLE {0 | OFF | 1 | ON}
```

```
TRACe<x>:EMC:RECALL:ENABLE?
```

## Arguments

0 or OFF indicates recalled trace is disabled 1 or ON indicates recalled trace is enabled

## Returns

<Boolean> 0,1

## Examples

TRACe4:EMC:RECALL:ENABLE might return 1, indicating that the recalled trace 4 is enabled (displayed).

## **TRACe<x>:EMC:RECAI:SElect**

Sets or queries the selection of specified recalled trace in live trace of EMC View.

<x> = <NR1> either 1,2,3,4,5 (trace number)

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Trace commands

### **Syntax**

TRACe<x>:EMC:RECAI:SElect

TRACe<x>:EMC:RECAI:SElect?

### **Arguments**

None

### **Returns**

<Boolean> 0,1

### **Examples**

TRACe3:EMC:RECAI:SElect might return 1, indicating that recalled trace 3 is selected.

## **TRACe<x>:EMC:SCAN:COUNT**

Sets or queries the count value of specified scan trace in EMC View.

<x> = <NR1> 1 or 2 (trace number)

### **Conditions**

Measurement view: EMC - EMI Display

### **Group**

Trace commands

### **Syntax**

TRACe<x>:EMC:SCAN:COUNT <value>

TRACe<x>:EMC:SCAN:COUNT?

### **Arguments**

<value> = <NR1>



## Returns

<NR1>

## Examples

TRACe1:EMC:SCAN:COUNT might return 12, indicating the count value for trace 1 is 12.

## TRACe<x>:EMC:SCAN:COUNT:RESet (No Query Form)

Resets the count value of the specified live scan trace in EMC view.

<x> = <NR1> 1 or 2 (trace number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Mass Memory commands

## Syntax

```
TRACe<x>:EMC:SCAN:COUNT:RESet
```

## Arguments

None

## Returns

None

## Examples

TRACe2:EMC:SCAN:COUNT:RESet resets the count value of live scan trace 2.

## TRACe<x>:EMC:SCAN:ENABLE

Sets or queries that selected scan trace is enabled (displayed) or disabled in EMC View.

<x> = <NR1> 1 or 2 (trace number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe<x>:EMC:SCAN:ENABLE { 0 | OFF | 1 | ON }
```

```
TRACe<x>:EMC:SCAN:ENABLE?
```

## Arguments

0 or OFF indicates the scan trace is disabled. 1 or ON indicates the scan trace is enabled.

## Returns

<Boolean> 0,1

## Examples

TRACe1:EMC:SCAN:ENABle? might return 1, indicating that the scan trace 1 is enabled (displayed).

## TRACe<x>:EMC:SCAN:FUNction

Sets or queries the scan function for the selected scan trace in EMC View.

<x> = <NR1>1 or 2 Trace number

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe<x>:EMC:SCAN:FUNction {NONE | MAXH | AVER | AVGL}
```

```
TRACe<x>:EMC:SCAN:FUNction?
```

## Arguments

NONE = Normal is the function selected MAXH = Max Hold is the function selected AVER = Avg(VRMS) is the function selected

AVGL = Avg (of logs) is the function selected

## Returns

<String> See Arguments

## Examples

TRACe1:EMC:SCAN:ENABle? might returns MAXH, indicating that the scan function selected for trace 1 is Max Hold.

## TRACe<x>:EMC:SCAN:SElect

Sets or queries the selection of scan trace in live trace of EMC View.

<x> = <NR1>1 or 2 (trace number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe<x>:EMC:SCAN:SElect
```

```
TRACe<x>:EMC:SCAN:SElect?
```

## Arguments

None

## Returns

<Boolean> 0,1

## Examples

TRACe1:EMC:SCAN:SElect might return 1, indicating that scan trace 1 is selected.

## TRACe<x>:EMC:SPOT:ENABLE

Sets or queries that selected spot trace is enabled (displayed) or disabled in EMC View.

<x> = <NR1> either 1 , 2 or 3 (trace number)

## Conditions

Measurement view: EMC - EMI Display

## Group

Trace commands

## Syntax

```
TRACe<x>:EMC:SPOT:ENABLE {0 | OFF | 1 | ON}
```

```
TRACe<x>:EMC:SPOT:ENABLE?
```

## Arguments

0 or OFF indicates spot trace is disabled 1 or ON indicates spot trace is enabled

## Returns

<Boolean> 0,1

## Examples

TRACe2:EMC:SPOT:ENABLE might return 1, indicating that the spot trace 2 is enabled (displayed).

## TRACe<x>:EMC:SPOT:SElect

Sets or queries the selection of spot trace in live trace of EMC View.

<x> = <NR1> either 1, 2 or 3 (trace number)

### Conditions

Measurement view: EMC - EMI Display

### Group

Trace commands

### Syntax

```
TRACe<x>:EMC:SPOT:SElect
```

```
TRACe<x>:EMC:SPOT:SElect?
```

### Arguments

None

### Returns

<Boolean> 0,1

### Examples

TRACe1:EMC:SPOT:SElect might return 1, indicating that spot trace 1 is selected.

## TRACe<x>:{FSETtling|PSETtling}:FREeze

Determines whether to freeze the specified trace in the Settling Time measurement display.

The parameter <x> = 1 or 2.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Trace commands

### Syntax

```
TRACe<x>:{FSETtling|PSETtling}:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:{FSETtling|PSETtling}:FREeze?
```

### Arguments

OFF or 0 updates the trace display normally.

ON or 1 stops updating trace display.

### Examples

TRACE1:FSETTLING:FREEZE ON stops updating the Trace 1 display.

## TRACe<x>:{FSETtling|PSETtling}:SElect

Selects the trace in the settling time measurement. The query returns the currently selected trace.

The parameter <x> = 1 or 2.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Trace commands

### Syntax

```
TRACe<x>:{FSETtling|PSETtling}:SElect
```

```
TRACe<x>:{FSETtling|PSETtling}:SElect?
```

### Arguments

None

### Returns

0 (not selected) or 1 (selected).

### Examples

```
TRACE2:FSETTLING:SELECT selects Trace 2.
```

## TRACe<x>:{FSETtling|PSETtling}:SHOW

Determines whether to show or hide the specified trace in the settling time view. The parameter <x> = 1 or 2.

### Conditions

Measurement views: Frequency and Phase Settling Time

### Group

Trace commands

### Syntax

```
TRACe<x>:{FSETtling|PSETtling}:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:{FSETtling|PSETtling}:SHOW?
```

### Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

## Examples

TRACe1 : PSETTLING : SHOW ON shows Trace 1 in the Phase Settling Time view.

## TRACe<x>:FVTime[:MEASview<y>]:DETection

Enables or queries the type of detection for the specified trace. The parameter <x> represents traces 1 to 4.

## Conditions

Measurement views: Frequency versus Time

## Group

Trace commands

## Syntax

```
TRACe<x>:FVTime[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative | POSNegative | SAMPlE }
```

```
TRACe<x>:FVTime[:MEASview<y>]:DETection?
```

## Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

AVERAge displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

## Examples

TRACe2:FVTime[:MEASview<y>]:DETection POSitive enables positive detection on Trace2.

## TRACe<x>:LTE:ACLR:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for multitrace functions (Max Hold and Average) in the LTE ACLR measurement. This command is effective when TRACe<x>: LTE:ACLR:FUNction is set to MAXHold or AVERAge and INITiate:CONTinuous is set to OFF.

## Conditions

Measurement view: LTE ACLR

## Group

Trace commands

## Syntax

```
TRACe<x>:LTE:ACLR:COUNT <number>
```

## Related Commands

```
TRACe<x>:LTE:ACLR:FUNCTION
```

## Arguments

<number>: : <NR1> specifies the count for multi-trace functions. Range: 1 to 10000.

## Returns

<NR1>

## Examples

```
TRACE1:LTE:ACLR:COUNT 32 sets the count to 32 for multitrace functions.
```

## TRACe<x>:LTE:ACLR:COUNT:RESet (No Query Form)

Clears the multifunction (Max Hold or Average) data and counter and restarts the process in the LTE ACLR display. This command is effective when `TRACe<x>:LTE:ACLR:FUNCTION` is set to `MAXHold` or `AVERAge`.

## Conditions

Measurement view: LTE ACLR

## Group

Trace commands

## Syntax

```
TRACe<x>:LTE:ACLR:COUNT:RESet
```

## Related Commands

```
TRACe<x>:LTE:ACLR:FUNCTION
```

## Arguments

None

## Examples

```
TRACE1:LTE:ACLR:COUNT:RESET clears the multifunction data and counter, and restarts the process.
```

## TRACe<x>:LTE:ACLR:FUNction

Sets or queries the trace function in the LTE ACLR display.

### Conditions

Measurement view: LTE ACLR

### Group

Trace commands

### Syntax

```
TRACe<x>:LTE:ACLR:FUNction { NONE | MAXH | AVER }
```

```
TRACe<x>:LTE:ACLR:FUNction?
```

### Arguments

NONE sets Normal display of Trace 1.

MAXHold sets Max Hold display of Trace 1, which indicates the maximum amplitude drift at each frequency point.

AVERage sets Average display of Trace 1, which indicates the average amplitude drift at each frequency point.

### Returns

NONE means that Trace 1 is set to Normal display. MAXHold means Trace 1 is set to Max Hold display. AVERage means Trace 1 is set to Average display.

### Examples

```
TRACE1:LTE:ACLR:FUNCTION MAXH displays the Max Hold trace in the LTE ACLR display.
```

## TRACe<x>:LTE:CONStE:SYNChronization:SEQUence:STATe? (Query Only)

Returns whether or not a measurement signal (Primary Synchronization Signal/Secondary Synchronization Signal) is selected for the specified trace in the Trace tab of the settings control panel of the LTE Constellation display.

### Conditions

Measurement view: LTE Constellation

### Group

Trace commands

### Syntax

```
TRACe<x>:LTE:CONStE:SYNChronization:SEQUence:STATe?
```

### Returns

0 means no measurement signal is selected.

1 means a measurement signal is selected.



## Examples

TRACE1:LTE:CONSTE:SYNCHRONIZATION:SEQUENCE:STATE? might return 1, indicating that SSS or PSS is selected in the Trace tab of the LTE Constellation display.

## TRACe<x>:NOISe:FIGure:AVERageCOUNT

Sets or queries the average counts for the specified trace in the Noise Figure display.

### Conditions

Measurement view: Noise Figure

The parameter <x>= 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:FIGure:AVERageCOUNT <NR1>
```

```
TRACe<x>:NOISe:FIGure:AVERageCOUNT?
```

### Arguments

<NR1> = integer

Range: 2 to 10000

### Returns

<NR3> floating point value with an exponent, which is the average count value.

## Examples

TRAC1:NOIS:FIG:AVER:COUN? might return 10.0000000000E+3, indicating that the average count value is set to 10000.

## TRACe<x>:NOISe:FIGure:COUNT:ENABLE

Sets or queries whether or not the count feature for the specified trace in the Noise Figure display is enabled or disabled.

### Conditions

Measurement view: Noise Figure display The parameter <x>= 1 to 3.

Only available when trace Function is set to Avg (VRMS), Max Hold, or Min Hold.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:FIGure:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

TRACe<x>:NOISe:FIGure:COUNT:ENABLE?

### Arguments

OFF or 0 disables the count feature for the specified trace.

ON or 1 enables the count feature for the specified trace.

### Returns

0 indicates the count feature for the specified trace if disabled.

1 indicates the count feature for the specified trace if enabled.

### Examples

TRAC1:NOIS:FIG:COUN:ENAB ON enables the count feature for Trace 1.

TRAC1:NOIS:FIG:COUN:ENAB? might return 0, indicating that the count feature for Trace 1 is disabled.

## TRACe<x>:NOISe:FIGure:DETection

Sets or queries the trace detection method in the Noise Figure display.

### Conditions

Measurement view: Noise Figure display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:FIGure:DETection { AVERage }
```

```
TRACe<x>:NOISe:FIGure:DETection?
```

### Arguments

**AVERage:** Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

### Returns

AVER means that the trace function is set to Average (VRMS).

### Examples

TRAC1:NOIS:FIG:DET AVER sets the detection method for Trace 1 to Average (VRMS).

TRACE2:NOIS:FIG:DET? might return AVER, indicating that the function for Trace 1 is set to Average (VRMS).

## TRACe<x>:NOISe:FIGure:FREeze

Specifies whether or not to freeze (halt updates of) the specified trace in the Noise Figure display.

### Conditions

Measurement view: Noise Figure display The parameter <x>= 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:FIGure:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:FIGure:FREeze?
```

### Arguments

OFF or 0 normally updates the display of the specified trace.

ON or 1 stops updating the display of the specified trace.

### Returns

0 indicates the specified trace is being updated normally in the Noise Figure display.

1 indicates the specified trace is not being updated in the Noise Figure display.

### Examples

```
TRAC1:NOIS:FIG:FRE ON freezes the display of Trace 1.
```

```
TRAC1:NOIS:FIG:FRE? might return 0, indicating that the display of Trace 1 is set to Freeze.
```

## TRACe<x>:NOISe:Figure:FUNCTion

Sets or queries the trace processing method (function) in the Noise Figure display.

### Conditions

Measurement view: Noise Figure display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:Figure:FUNCTion { NORMAl | AVERAge | MAXHold| MINHold}
```

```
TRACe<x>:NOISe:Figure:FUNCTion?
```

### Arguments

NORMAl: Displays the trace record for each display point without additional processing.

**AVERage:** Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

**MAXHold:** Displays the maximum value in the trace record for each display point.

**MINHold:** Displays the minimum value in the trace record for each display point.

## Returns

NORM means that the trace function is set to Normal.

AVER means that the trace function is set to Average (VRMS).

MAXH means that the trace function is set to Max Hold.

MINH means that the trace function is set to Min Hold.

## Examples

TRAC1:NOIS:FIG:FUNC NORM sets the function for Trace 1 to Normal.

TRAC2:NOIS:FIG:FUNC? might return MAXH, indicating that the function for Trace 1 is set to Max Hold.

## TRACe<x>:NOISe:FIGure:SElect

Sets the trace to be shown in the Noise Figure display. The query form of this command returns whether or not the specified trace is set to show (ON) or not (OFF).

## Conditions

Measurement view: Noise Figure display The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

TRACe<x>:NOISe:FIGure:SElect

TRACe<x>:NOISe:FIGure:SElect?

## Arguments

None

## Returns

0 means that the specified trace is not shown (set to Off).

1 means that the specified trace is set to show (set to normal).

## Examples

TRAC1:NOIS:FIG:SEL will set the display to show Trace 1.

TRAC2:NOIS:FIG:SEL? might return 1, indicating that Trace 2 is set to show.

## TRACe<x>:NOISe:FIGure:SHOW

Sets or queries the state (shown or not shown) of the specified trace in the Noise Figure display.

### Conditions

Measurement view: Noise Figure display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:FIGure:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:FIGure:SHOW?
```

### Arguments

OFF or 0 hides the specified trace.

ON or 1 displays the specified trace.

### Returns

0 indicates the specified trace is not shown in the Noise Figure display.

1 indicates the specified trace is shown in the Noise Figure display.

### Examples

TRAC1:NOIS:FIG:SHOW sets Trace to show in the Noise Figure display.

TRAC2:NOIS:FIG:SHOW? might return 0, indicating that Trace 2 is not showing in the Noise Figure display.

## TRACe<x>:NOISe:GAIN:AVERAge:COUNT

Sets or queries the average counts for the specified trace in the Gain display.

### Conditions

Measurement view: Gain

The parameter <x>= 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:GAIN:AVERAge:COUNT <NR1>
```

```
TRACe<x>:NOISe:GAIN:AVERAge:COUNT?
```

## Arguments

<NR1> = integer

Range: 2 to 10000

## Returns

<NR3> floating point value with an exponent, which is the average count value.

## Examples

TRAC1:NOIS:GAIN:AVER:COUN? might return 10.000000000E+3, indicating that the average count value is set to 10000.

## TRACe<x>:NOISe:GAIN:COUNT:ENABLE

Sets or queries whether or not the count feature for the specified trace in the Gain display is enabled or disabled.

## Conditions

Measurement view: Gain display

The parameter <x>= 1 to 3.

Only available when trace Function is set to Avg (VRMS), Max Hold, or Min Hold.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:GAIN:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:GAIN:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count feature for the specified trace.

ON or 1 enables the count feature for the specified trace.

## Returns

0 indicates the count feature for the specified trace if disabled.

1 indicates the count feature for the specified trace if enabled.

## Examples

TRAC1:NOIS:GAIN:COUN:ENAB ON enables the count feature for Trace 1.

TRAC1:NOIS:GAIN:COUN:ENAB? might return 0, indicating that the count feature for Trace 1 is disabled.

## TRACe<x>:NOISe:GAIN:DETection

Sets or queries the trace detection method in the Gain display.

## Conditions

Measurement view: Gain display

The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:GAIN:DETection { AVERage }
```

```
TRACe<x>:NOISe:GAIN:DETection?
```

## Arguments

**AVERage:** Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

## Returns

AVER means that the trace function is set to Average (VRMS).

## Examples

```
TRACE1:NOIS:YFAC:DET AVER sets the detection method for Trace 1 to Average (VRMS).
```

```
TRACE2:NOIS:YFAC:DET? might return AVER, indicating that the function for Trace 1 is set to Average (VRMS).
```

## TRACe<x>:NOISe:GAIN:FRoze

Specifies whether or not to freeze (halt updates of) the specified trace in the Gain display.

## Conditions

Measurement view: Gain display

The parameter <x>= 1 to 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:GAIN:FRoze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:GAIN:FRoze?
```

## Arguments

OFF or 0 normally updates the display of the specified trace.

ON or 1 stops updating the display of the specified trace.

## Returns

0 indicates the specified trace is being updated normally in the Gain display.

1 indicates the specified trace is not being updated in the Gain display.

## Examples

`TRAC1:NOIS:GAIN:FRE ON` freezes the display of Trace 1.

`TRAC1:NOIS:GAIN:FRE?` might return 0, indicating that the display of Trace 1 is set to Freeze.

## TRACe<x>:NOISe:GAIN:FUNCTion

Sets or queries the trace processing method (function) in the Gain display.

## Conditions

Measurement view: Gain display

The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

`TRACe<x>:NOISe:GAIN:FUNCTion { NORMAl | AVERAge | MAXHold| MINHold}`

`TRACe<x>:NOISe:GAIN:FUNCTion?`

## Arguments

**NORMAl**: Displays the trace record for each display point without additional processing.

**AVERAge**: Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

**MAXHold**: Displays the maximum value in the trace record for each display point.

**MINHold**: Displays the minimum value in the trace record for each display point.

## Returns

**NORM** means that the trace function is set to Normal.

**AVER** means that the trace function is set to Average (VRMS).

**MAXH** means that the trace function is set to Max Hold.

**MINH** means that the trace function is set to Min Hold.

## Examples

`TRAC1:NOIS:GAIN:FUNC NORM` sets the function for Trace 1 to Normal.



TRAC2:NOIS:GAIN:FUNC? might return MAXH, indicating that the function for Trace 1 is set to Max Hold.

## TRACe<x>:NOISe:GAIN:SElect

Sets the trace to be shown in the Gain display. The query form of this command returns whether or not the specified trace is set to show (ON) or not (OFF).

### Conditions

Measurement view: Gain display

The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:GAIN:SElect
```

```
TRACe<x>:NOISe:GAIN:SElect?
```

### Arguments

None

### Returns

0 means that the specified trace is not shown (set to Off).

1 means that the specified trace is set to show (set to normal).

### Examples

TRAC1:NOIS:GAIN:SEL will set the display to show Trace 1.

TRAC2:NOIS:GAIN:SEL? might return 1, indicating that Trace 2 is set to show.

## TRACe<x>:NOISe:GAIN:SHOW

Sets or queries the state (shown or not shown) of the specified trace in the Gain display.

### Conditions

Measurement view: Gain display

The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:GAIN:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:GAIN:SHOW?
```

## Arguments

OFF or 0 hides the specified trace.

ON or 1 displays the specified trace.

## Returns

0 indicates the specified trace is not shown in the Gain display.

1 indicates the specified trace is shown in the Gain display.

## Examples

TRAC1:NOIS:GAIN:SHOW sets Trace to show in the Gain display.

TRAC2:NOIS:GAIN:SHOW? might return 0, indicating that Trace 2 is not showing in the Gain display.

## TRACe<x>:NOISe:TEMPerature:AVERageCOUNT

Sets or queries the average counts for the specified trace in the Noise Temperature display.

## Conditions

Measurement view: Noise Temperature The parameter <x>= 1 to 3.

## Group

Trace commands

## Syntax

TRACe<x>:NOISe:TEMPerature:AVERageCOUNT <NR1>

TRACe<x>:NOISe:TEMPerature:AVERageCOUNT?

## Arguments

<NR1> = integer

Range: 2 to 10000

## Returns

<NR3> floating point value with an exponent, which is the average count value.

## Examples

TRAC1:NOIS:TEMP:AVER:COUN? might return 10.000000000E+3, indicating that the average count value is set to 10000.

## TRACe<x>:NOISe:TEMPerature:COUNT:ENABLE

Sets or queries whether or not the count feature for the specified trace in the Noise Temperature display is enabled or disabled.

## Conditions

Measurement view: Noise Temperature display The parameter <x>= 1 to 3.

Only available when trace Function is set to Avg (VRMS), Max Hold, or Min Hold.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:TEMPeRature:COUNT:ENABle { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:TEMPeRature:COUNT:ENABle?
```

## Arguments

OFF or 0 disables the count feature for the specified trace.

ON or 1 enables the count feature for the specified trace.

## Returns

0 indicates the count feature for the specified trace if disabled.

1 indicates the count feature for the specified trace if enabled.

## Examples

```
TRAC1:NOIS:TEMP:COUN:ENAB ON enables the count feature for Trace 1.
```

```
TRAC1:NOIS:TEMP:COUN:ENAB? might return 0, indicating that the count feature for Trace 1 is disabled.
```

## TRACe<x>:NOISe:TEMPeRature:DETection

Sets or queries the trace detection method in the Noise Temperature display.

## Conditions

Measurement view: Noise Temperature display The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:TEMPeRature:DETection { AVERage }
```

```
TRACe<x>:NOISe:TEMPeRature:DETection?
```

## Arguments

**AVERage:** Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

## Returns

AVER means that the trace function is set to Average (VRMS).

## Examples

TRAC1:NOIS:TEMP:DET AVER sets the detection method for Trace 1 to Average (VRMS).

TRAC2:NOIS:TEMP:DET? might return AVER, indicating that the function for Trace 1 is set to Average (VRMS).

## TRACe<x>:NOISE:TEMPerature:FREeze

Specifies whether or not to freeze (halt updates of) the specified trace in the Noise Temperature display.

### Conditions

Measurement view: Noise Temperature display The parameter <x>= 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISE:TEMPerature:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISE:TEMPerature:FREeze?
```

### Arguments

OFF or 0 normally updates the display of the specified trace.

ON or 1 stops updating the display of the specified trace.

### Returns

0 indicates the specified trace is being updated normally in the Noise Temperature display.

1 indicates the specified trace is not being updated in the Noise Temperature display.

## Examples

TRAC1:NOIS:TEMP:FRE ON freezes the display of Trace 1.

TRAC1:NOIS:TEMP:FRE? might return 0, indicating that the display of Trace 1 is set to Freeze.

## TRACe<x>:NOISe:TEMPerature:FUNCTION

Sets or queries the trace processing method (function) in the Noise Temperature display.

### Conditions

Measurement view: Noise Temperature display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:TEMPerature:FUNCTION { NORMal | AVERage | MAXHold | MINHold }
```

TRACe<x>:NOISe:TEMPerature:FUNCTion?

## Arguments

**NORMa1**: Displays the trace record for each display point without additional processing.

**AVERage**: Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

**MAXHold**: Displays the maximum value in the trace record for each display point.

**MINHold**: Displays the minimum value in the trace record for each display point.

## Returns

**NORM** means that the trace function is set to Normal.

**AVER** means that the trace function is set to Average (VRMS).

**MAXH** means that the trace function is set to Max Hold.

**MINH** means that the trace function is set to Min Hold.

## Examples

TRAC1:NOIS:TEMP:FUNC NORM sets the function for Trace 1 to Normal.

TRAC2:NOIS:TEMP:FUNC? might return MAXH, indicating that the function for Trace 1 is set to Max Hold.

## TRACe<x>:NOISe:TEMPerature:SElect

Sets the trace to be shown in the Noise Temperature display. The query form of this command returns whether or not the specified trace is set to show (On) or not (Off).

## Conditions

Measurement view: Noise Temperature display The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

TRACe<x>:NOISe:TEMPerature:SElect

TRACe<x>:NOISe:TEMPerature:SElect?

## Arguments

None

## Returns

0 means that the specified trace is not shown (set to Off).

1 means that the specified trace is set to show (set to normal).

## Examples

TRAC1:NOIS:TEMP:SEL will set the display to show Trace 1.

TRAC2:NOIS:TEMP:SEL? might return 1, indicating that Trace 2 is set to show.

## TRACe<x>:NOISe:TEMPerature:SHOW

Sets or queries the state (shown or not shown) of the specified trace in the Noise Temperature display.

### Conditions

Measurement view: Noise Temperature display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:TEMPerature:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:TEMPerature:SHOW?
```

### Arguments

OFF or 0 hides the specified trace.

ON or 1 displays the specified trace.

### Returns

0 indicates the specified trace is not shown in the Noise Temperature display.

1 indicates the specified trace is shown in the Noise Temperature display.

## Examples

TRAC1:NOIS:TEMP:SHOW sets Trace to show in the Noise Temperature display.

TRAC2:NOIS:TEMP:SHOW? might return 0, indicating that Trace 2 is not showing in the Noise Temperature display.

## TRACe<x>:NOISe:YFACTOR:AVERAge:COUNT

Sets or queries the average counts for the specified trace in the Y Factor display.

### Conditions

Measurement view: Y Factor

The parameter <x>= 1 to 3.

### Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:YFACTOR:AVERage:COUNT <NR1>
```

```
TRACe<x>:NOISe:YFACTOR:AVERage:COUNT?
```

## Arguments

<NR1> = integer

Range: 2 to 10000

## Returns

<NR3> floating point value with an exponent, which is the average count value.

## Examples

TRAC1:NOIS:TEMP:AVER:COUN? might return 10.000000000E+3, indicating that the average count value is set to 10000.

## TRACe<x>:NOISe:YFACTOR:COUNT:ENABLE

Sets or queries whether or not the count feature for the specified trace in the Y Factor display is enabled or disabled.

## Conditions

Measurement view: Y Factor display The parameter <x>= 1 to 3.

Only available when trace Function is set to Avg (VRMS), Max Hold, or Min Hold.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:YFACTOR:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:YFACTOR:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count feature for the specified trace.

ON or 1 enables the count feature for the specified trace.

## Returns

0 indicates the count feature for the specified trace if disabled.

1 indicates the count feature for the specified trace if enabled.

## Examples

TRAC1:NOIS:YFAC:COUN:ENAB ON enables the count feature for Trace 1.

TRAC1:NOIS:YFAC:COUN:ENAB? might return 0, indicating that the count feature for Trace 1 is disabled.

## TRACe<x>:NOISe:YFACtor:DETection

Sets or queries the trace detection method in the Y Factor display.

### Conditions

Measurement view: Y Factor display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:YFACtor:DETection { AVERage }
```

```
TRACe<x>:NOISe:YFACtor:DETection?
```

### Arguments

**AVERage:** Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

### Returns

AVER means that the trace function is set to Average (VRMS).

### Examples

```
TRACE1:NOIS:YFAC:DET AVER sets the detection method for Trace 1 to Average (VRMS).
```

```
TRACE2:NOIS:YFAC:DET? might return AVER, indicating that the function for Trace 1 is set to Average (VRMS).
```

## TRACe<x>:NOISE:YFACtor:FREEze

Specifies whether or not to freeze (halt updates of) the specified trace in the Y Factor display.

### Conditions

Measurement view: Y Factor display The parameter <x>= 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISE:YFACtor:FREEze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISE:YFACtor:FREEze?
```

### Arguments

OFF or 0 normally updates the display of the specified trace.



ON or 1 stops updating the display of the specified trace.

## Returns

0 indicates the specified trace is being updated normally in the Y Factor display.

1 indicates the specified trace is not being updated in the Y Factor display.

## Examples

TRAC1:NOIS:YFAC:FRE ON freezes the display of Trace 1.

TRAC1:NOIS:YFAC:FRE? might return 0, indicating that the display of Trace 1 is set to Freeze.

## TRACe<x>:NOISe:YFACtor:FUNCTion

Sets or queries the trace processing method (function) in the Y Factor display.

## Conditions

Measurement view: Y Factor display The parameter <x> = 1 to 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:NOISe:YFACtor:FUNCTion { NORMAl | AVERAge | MAXHold| MINHold}
```

```
TRACe<x>:NOISe:YFACtor:FUNCTion?
```

## Arguments

**NORMAl**: Displays the trace record for each display point without additional processing.

**AVERAge**: Each point on the trace is the result of determining the RMS Voltage value for all of the IQ samples available to the trace point. When displayed in either linear (Volts, Watts) or Log (dB, dBm), the correct RMS value results.

When the averaging function is applied to a trace, the averaging is performed on the linear (Voltage) values, resulting in the correct average for RMS values.

**MAXHold**: Displays the maximum value in the trace record for each display point.

**MINHold**: Displays the minimum value in the trace record for each display point.

## Returns

NORM means that the trace function is set to Normal.

AVER means that the trace function is set to Average (VRMS).

MAXH means that the trace function is set to Max Hold.

MINH means that the trace function is set to Min Hold.

## Examples

TRAC1:NOIS:YFAC:FUNC NORM sets the function for Trace 1 to Normal.

TRAC2:NOIS:YFAC:FUNC? might return MAXH, indicating that the function for Trace 1 is set to Max Hold.

## TRACe<x>:NOISe:YFACtor:SElect

Sets the trace to be shown in the Y Factor display. The query form of this command returns whether or not the specified trace is set to show (ON) or not (OFF).

### Conditions

Measurement view: Y Factor display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:YFACtor:SElect
```

```
TRACe<x>:NOISe:YFACtor:SElect?
```

### Arguments

None

### Returns

0 means that the specified trace is not shown (set to Off).

1 means that the specified trace is set to show (set to normal).

### Examples

TRAC1:NOIS:YFAC:SEL will set the display to show Trace 1.

TRAC2:NOIS:YFAC:SEL? might return 1, indicating that Trace 2 is set to show.

## TRACe<x>:NOISe:YFACtor:SHOW

Sets or queries the state (shown or not shown) of the specified trace in the Y Factor display.

### Conditions

Measurement view: Y Factor display The parameter <x> = 1 to 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:NOISe:YFACtor:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:NOISe:YFACtor:SHOW?
```

## Arguments

OFF or 0 hides the specified trace.

ON or 1 displays the specified trace.

## Returns

0 indicates the specified trace is not shown in the Y Factor display.

1 indicates the specified trace is shown in the Y Factor display.

## Examples

`TRACE1:NOISE:YFACTOR:SHOW` sets Trace to show in the Y Factor display.

`TRACE2:NOISE:YFACTOR:SHOW?` might return 0, indicating that Trace 2 is not showing in the Y Factor display.

## TRACe<x>:OFDM:EVM

Determines whether or not to show the specified trace in the OFDM EVM view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:EVM { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:EVM?
```

## Arguments

OFF or 0 hides the specified trace in the view.

ON or 1 shows the specified trace in the view.

## Examples

`TRACe1:OFDM:EVM ON` shows Trace 1 in the OFDM EVM view.

## TRACe<x>:OFDM:EVM:FREEze

Determines whether or not to freeze the display of the specified trace in the OFDM EVM view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:EVM:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:EVM:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

## Examples

```
TRACE1:OFDM:EVM:FREEZE ON freezes the display for Trace 1.
```

## TRACe<x>:OFDM:EVM:SElect

Sets or queries the specified trace to display in the OFDM EVM view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:EVM:SElect
```

```
TRACe<x>:OFDM:EVM:SElect?
```

## Arguments

None

## Returns

```
{ OFF | ON | 0 | 1 }
```

OFF or 0 indicates that the readout is not being displayed for the specified trace.

ON or 1 indicates that the readout is being displayed for the specified trace.

## Examples

```
TRACE1:OFDM:EVM:SELECT selects Trace 1 (Average trace) to display the readout.
```

```
TRACE1:OFDM:EVM:SELECT? might return 1, indicating that the readout is being displayed for Trace 1.
```

## TRACe<x>:OFDM:MERRor

Determines whether or not to show the specified trace in the OFDM Magnitude error view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe<x>:OFDM:MERRor { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:MERRor?
```

### Arguments

OFF or 0 hides the specified trace in the view.

ON or 1 shows the specified trace in the view.

### Examples

```
TRACe1:OFDM:MERRor ON shows Trace 1 in the OFDM Magnitude error view.
```

## TRACe<x>:OFDM:MERRor:FREeze

Determines whether or not to freeze the display of the specified trace in the OFDM Magnitude error view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe<x>:OFDM:MERRor:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:MERRor:FREeze?
```

### Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

## Examples

TRACE1:OFDM:MERRor:FREEZE ON freezes the display for Trace 1.

## TRACe<x>:OFDM:MERRor:SElect

Sets or queries the specified trace to display in the OFDM Magnitude error view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe<x>:OFDM:MERRor:SElect { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:MERRor:SElect?
```

### Arguments

None

### Returns

```
{ OFF | ON | 0 | 1 }
```

OFF or 0 indicates that the readout is not being displayed for the specified trace.

ON or 1 indicates that the readout is being displayed for the specified trace.

## Examples

TRACE1:OFDM:MERRor:SElect selects Trace 1 (Average trace) to display the readout.

TRACE1:OFDM:MERRor:SElect? might return 1, indicating that the readout is being displayed for Trace 1.

## TRACe<x>:OFDM:PERRor

Determines whether or not to show the specified trace in the OFDM Phase error view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:PERRor { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:PERRor?
```

## Arguments

OFF or 0 hides the specified trace in the view.

ON or 1 shows the specified trace in the view.

## Examples

```
TRACe1:OFDM:PERRor ON shows Trace 1 in the OFDM Phase error view.
```

## TRACe<x>:OFDM:PERRor:FREeze

Determines whether or not to freeze the display of the specified trace in the OFDM Phase error view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:PERRor:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:PERRor:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

## Examples

```
TRACe1:OFDM:PERRor:FREeze ON freezes the display for Trace 1.
```

## TRACe<x>:OFDM:PERRor:SElect

Sets or queries the specified trace to display in the OFDM Phase error view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:PERROr:SElect { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:PERROr:SElect?
```

## Arguments

None

## Returns

```
{ OFF | ON | 0 | 1 }
```

OFF or 0 indicates that the readout is not being displayed for the specified trace.

ON or 1 indicates that the readout is being displayed for the specified trace.

## Examples

TRACE1:OFDM:PERROr:SElect selects Trace 1 (Average trace) to display the readout.

TRACE1:OFDM:PERROr:SElect? might return 1, indicating that the readout is being displayed for Trace 1.

## TRACe<x>:OFDM:POWer

Determines whether or not to show the specified trace in the OFDM Power view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

## Conditions

Measurement views: OFDM

## Group

Trace commands

## Syntax

```
TRACe<x>:OFDM:POWer { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:POWer?
```

## Arguments

OFF or 0 hides the specified trace in the view.

ON or 1 shows the specified trace in the view.

## Examples

TRACe1:OFDM:POWer ON shows Trace 1 in the OFDM Power view.



## TRACe<x>:OFDM:POWer:FREeze

Determines whether or not to freeze the display of the specified trace in the OFDM Power view.

The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe<x>:OFDM:POWer:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:POWer:FREeze?
```

### Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

### Examples

```
TRACE1:OFDM:POWer:FREeze ON freezes the display for Trace 1.
```

## TRACe<x>:OFDM:POWer:SELEct

Sets or queries the specified trace to display in the OFDM Power view. The parameter <x> = 1 or 2, where 1 is Average and 2 is Value.

### Conditions

Measurement views: OFDM

### Group

Trace commands

### Syntax

```
TRACe<x>:OFDM:POWer:SELEct { OFF | ON | 0 | 1 }
```

```
TRACe<x>:OFDM:POWer:SELEct?
```

### Arguments

None

### Returns

```
{ OFF | ON | 0 | 1 }
```

OFF or 0 indicates that the readout is not being displayed for the specified trace.

ON or 1 indicates that the readout is being displayed for the specified trace.

## Examples

TRACE1:OFDM:POWer:SELECT selects Trace 1 (Average trace) to display the readout.

TRACE1:OFDM:POWer:SELECT? might return 1, indicating that the readout is being displayed for Trace 1.

## TRACe<x>:P25:CONStellation

Specifies whether or not to show the specified trace (Trace 1 or 2) in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation display The parameter <x>= 1 or 2.

### Group

Trace commands

### Syntax

```
TRACe<x>:P25:CONStellation { OFF | ON | 0 | 1 }
```

```
TRACe<x>:P25:CONStellation?
```

### Arguments

OFF or 0 hides the specified trace.

ON or 1 displays the specified trace.

### Returns

0 indicates the specified trace is not shown in the P25 Constellation display.

1 indicates the specified trace is shown in the P25 Constellation display.

## Examples

TRAC1:P25:CONS ON specifies that Trace 1 be shown in the P25 Constellation display.

## TRACe<x>:P25:CONStellation:FREeze

Specifies whether or not to freeze (halt updates of) the specified trace (Trace 1 or 2) in the P25 Constellation display.

### Conditions

Measurement view: P25 Constellation display The parameter <x>= 1 or 2.

### Group

Trace commands

### Syntax

```
TRACe<x>:P25:CONStellation:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:P25:CONStellation:FREeze?
```

## Arguments

OFF or 0 normally updates the display of the specified trace.

ON or 1 stops updating the display of the specified trace.

## Returns

0 indicates the specified trace is being updated normally in the P25 Constellation display.

1 indicates the specified trace is not being updated in the P25 Constellation display.

## Examples

TRAC1 : P25 : CONS : FRE ON freezes the display of Trace 1.

TRAC1 : P25 : CONS : FRE? might return 0, indicating that the display of Trace 1 is set to Freeze.

## TRACe<x>:P25:CONStellation:MODE

Sets or queries how to display the trace content in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation display The parameter <x>= 1 or 2.

## Group

Trace commands

## Syntax

```
TRACe<x>:P25:CONStellation:MODE { VECT | SYMB | LIN }
```

## Arguments

VECT (vectors) connects adjacent symbol points with the signal locus.

SYMB (symbols) displays individual symbol points.

LIN (line) displays all symbol points as a line.

## Returns

VECT adjacent symbol points are connected with the signal locus.

SYMB individual symbol points are displayed.

LIN all symbol points are displayed as a line.

## Examples

TRAC1 : P25 : CONS : MODE SYMB sets the display to show the constellation of Trace 1 connecting adjacent symbol points with the signal locus.

## TRACe<x>:P25:CONStellation:SELEct

Selects the specified trace (Trace 1 or 2) in the P25 Constellation display.

## Conditions

Measurement view: P25 Constellation display The parameter <x>= 1 or 2.

## Group

Trace commands

## Syntax

```
TRACe<x>:P25:CONStellation:SElect { OFF | ON | 0 | 1 }
```

```
TRACe<x>:P25:CONStellation:SElect?
```

## Arguments

None

## Returns

0 indicates the specified trace is not the active trace in the P25 Constellation display.

1 indicates the specified trace is the active trace in the P25 Constellation display.

## Examples

TRAC1:P25:CONS:SEL? might return 1, indicating that Trace 1 is the selected trace in the P25 Constellation display.

## TRACe<x>:P25:EDIagram:ENABLE:I

Determines whether to show or hide the I trace in the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Trace commands

## Syntax

```
TRACe<x>:P25:EDIagram:ENABLE:I
```

```
TRACe<x>:P25:EDIagram:ENABLE:I?
```

## Related Commands

```
TRACe<x>:P25:EDIagram:ENABLE:Q
```

## Arguments

OFF or 0 hides the I trace.

ON or 1 shows the I trace.

## Returns

0 indicates the trace is off.

1 indicates the trace is on.

## Examples

`TRACEX:P25:EDIAGRAM:ENABLE:I ON` shows the I trace in the P25 Eye Diagram display.

## **TRACe<x>:P25:EDIagram:ENABLE:Q**

Determines whether to show or hide the Q trace in the P25 Eye Diagram display.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Trace commands

## Syntax

`TRACe<x>:P25:EDIagram:ENABLE:Q`

`TRACe<x>:P25:EDIagram:ENABLE:Q?`

## Related Commands

`TRACe<x>:P25:EDIagram:ENABLE:I`

## Arguments

OFF or 0 hides the Q trace.

ON or 1 shows the Q trace.

## Returns

0 indicates the trace is off.

1 indicates the trace is on.

## Examples

`TRACEX:P25:EDIAGRAM:ENABLE:Q ON` shows the Q trace in the P25 Eye Diagram display.

## **TRACe<x>:P25:EDIagram:SElect:I**

This command selects the I trace in the P25 Eye Diagram display. The query version of this command returns whether the I trace is selected or not.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Trace commands

## Syntax

```
TRACe<x>:P25:EDIagram:SElect:I
```

```
TRACe<x>:P25:EDIagram:SElect:I?
```

## Related Commands

```
TRACe<x>:P25:EDIagram:SElect:Q
```

## Arguments

None.

## Returns

0 indicates that the I trace is not the selected trace.

1 indicates that the I trace is the selected trace.

## Examples

TRACEX:P25:EDIAGRAM:SELECT:I selects the I trace in the P25 Eye Diagram display.

## TRACe<x>:P25:EDIagram:SElect:Q

This command selects the Q trace in the P25 Eye Diagram display. The query version of this command returns whether the Q trace is selected or not.

## Conditions

Measurement view: P25 Eye Diagram

## Group

Trace commands

## Syntax

```
TRACe<x>:P25:EDIagram:SElect:Q
```

```
TRACe<x>:P25:EDIagram:SElect:Q?
```

## Related Commands

```
TRACe<x>:P25:EDIagram:SElect:I
```

## Arguments

None.

## Returns

0 indicates that the Q trace is not the selected trace.

1 indicates that the Q trace is the selected trace.

## Examples

`TRACEX:P25:EDIAGRAM:SELECT:Q` selects the Q trace in the P25 Eye Diagram display.

## TRACe<x>:PHVTime[:MEASview<y>]:DETection

Sets or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

### Conditions

Measurement views: Phase versus Time

### Group

Trace commands

### Syntax

```
TRACe<x>:PHVTime[:MEASview<y>]:DETection { AVERage | POSitive | NEGative |
POSNegative | SAMPlE }
```

```
TRACe<x>:PHVTime[:MEASview<y>]:DETection?
```

### Arguments

<x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

<y> = 1 to 8; represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional. Default value 1.

AVERage displays the average data value for each pixel.

POSitive displays the maximum data value for each pixel.

NEGative displays the minimum data value for each pixel.

POSNegative displays the maximum and minimum data values for each pixel.

SAMPlE displays the first sample value received for each pixel.

### Examples

`TRACe1:PHVTime[:MEASview<y>]:DETection POSitive` displays the maximum data value for each pixel.

## TRACe<x>:PNOise:SElect

Selects the trace in the phase noise measurement. The query returns the currently selected trace.

The parameter <x> = 1 and 2.

### Conditions

Measurement views: Phase noise

## Group

Trace commands

## Syntax

```
TRACe<x>:PNOise:SElect
```

```
TRACe<x>:PNOise:SElect?
```

## Arguments

None

## Returns

0 (not selected) or 1 (selected).

## Examples

```
TRACE2:PNOISE:SELECT selects Trace 2.
```

## TRACe<x>:PNOise:SHOW

Shows or hides the specified trace in the phase noise view. The parameter <x> = 1 and 2.

## Conditions

Measurement views: Phase noise

## Group

Trace commands

## Syntax

```
TRACe<x>:PNOise:SHOW { OFF | ON | 0 | 1 }
```

```
TRACe<x>:PNOise:SHOW?
```

## Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

## Examples

```
TRACE1:PNOISE:SHOW ON shows Trace 1 in the phase noise view.
```

## TRACe<x>:PNOise:SMOothing:COUNT

Sets or queries the number of data points to take the moving average for smoothing the trace. This command is effective when `TRACe<x>:PNOise:SMOothing:ENABLE` is set to ON.

The parameter <x> = 1 and 2.



## Conditions

Measurement views: Phase noise

## Group

Trace commands

## Syntax

```
TRACe<x>:PNOise:SMOothing:COUNT <number>
```

```
TRACe<x>:PNOise:SMOothing:COUNT?
```

## Arguments

<number>: : <NR1> specifies the number of data points to take the moving average for smoothing. Range: 3 to 50.

## Examples

```
TRACE1:PNOISE:SMOOTHING:COUNT 16 sets the smoothing count to 16 for Trace 1.
```

## TRACe<x>:PNOise:SMOothing:ENABLE

Determines whether to enable or disable smoothing the specified trace in the phase noise view.

The parameter <x> = 1 and 2.

## Conditions

Measurement views: Phase noise

## Group

Trace commands

## Syntax

```
TRACe<x>:PNOise:SMOothing:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:PNOise:SMOothing:ENABLE?
```

## Arguments

OFF or 0 disables smoothing.

ON or 1 enables smoothing.

## Examples

```
TRACE1:PNOISE:SMOOTHING:ENABLE ON enables smoothing Trace 1 in the phase noise view.
```

## TRACe<x>:PNOise:SMOothing:RESet (No Query Form)

Restarts the smoothing process, discarding accumulated data and resetting the counter. This command is effective when TRACe<x>:PNOise:SMOothing: ENABLE is set to ON.

The parameter <x> = 1 and 2.

### Conditions

Measurement views: Phase noise

### Group

Trace commands

### Syntax

```
TRACe<x>:PNOise:SMOothing:RESet
```

### Arguments

None

### Examples

TRACE1:PNOISE:SMOOTHING:RESET restarts the smoothing process for Trace 1.

## TRACe<x>:SPECTrum[:MEASview<y>]

Sets or queries the Reference level of measurement source for Q data type.

### Conditions

Measurement views: Spectrum

### Group

Trace commands

### Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>] { OFF | ON | 0 | 1 }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]?
```

### Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

For Trace 5 (spectrogram), use the TRACe:SGRam:SElect:LINE command to select the line to send to the spectrum display.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

TRACE1:SPECTRUM:MEASview2 ON shows Trace 1 in the Spectrum Analyzer view.

## TRACe<x>:SPECTrum[:MEASview<y>]:AVERAge:COUNT

Sets or queries the number of traces to combine. This command is effective when TRACe<x>:SPECTrum[:MEASview<y>]:FUNctioN is AVERAge, MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

### Conditions

Measurement views: Spectrum

### Group

Trace commands

### Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:AVERAge:COUNT <number>
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:AVERAge:COUNT?
```

### Arguments

<number>: : <NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

### Examples

```
TRACE1 : SPECTRUM : MEASVIEW2 : AVERAGE : COUNT 64 sets the average count to 64 for Trace 1.
```

## TRACe<x>:SPECTrum[:MEASview<y>]:AVERAge:RESet (No Query Form)

Clears average data and counter, and restarts the average process for the specified trace in the Spectrum view. This command is effective when TRACe<x>:SPECTrum[:MEASview<y>]:FUNctioN is set to AVERAge, MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

### Conditions

Measurement views: Spectrum

### Group

Trace commands

### Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:AVERAge:RESet
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1 : SPECTRUM : MEASVIEW2 : AVERAGE : RESET clears average data and counter, and restarts the average process for Trace 1.

## TRACe<x>:SPECTrum[:MEASview<y>]:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Max or Min Hold trace in the Spectrum measurement. This command is effective when TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold and INITiate:CONTinuous is set to OFF.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:COUNT <number>
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:COUNT?
```

## Arguments

<number>: : <NR1> specifies the count for Max/Min Hold.

Range: 1 to 10000.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1 : SPECTRUM : MEASVIEW2 : COUNT 32 sets the count to 32 for Trace 1.

## TRACe<x>:SPECTrum[:MEASview<y>]:COUNT:ENABLE

Determines whether to enable or disable the count for the Max or Min Hold trace in the Spectrum view. This command is effective when TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count for the Max/Min Hold trace.

ON or 1 enables the count for the Max/Min Hold trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
TRACe1:SPECTRUM:MEASVIEW2:COUNT:ENABLE ON enables the Max/Min Hold count for Trace 1.
```

## TRACe<x>:SPECTrum[:MEASview<y>]:COUNT:RESet (No Query Form)

Clears the Max or Min Hold data and counter, and restarts the process for the specified trace in the Spectrum view. This command is effective when TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION is set to MAXHold or MINHold.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:COUNT:RESet
```

## Arguments

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

`TRACE1 : SPECTRUM : MEASVIEW2 : COUNT : RESET` clears the Max/Min Hold data and counter, and restarts the process for Trace 1.

## TRACe<x>:SPECTrum[:MEASview<y>]:DETection

Sets or queries the display detector (method to be used for decimating traces to fit the available horizontal space on screen). The number of horizontal pixels on screen is generally smaller than that of waveform data points. When actually displayed, the waveform data is therefore thinned out, according to the number of pixels, for being compressed.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:DETection { AVERAge | POSitive | NEGative | CAVerage
| CPEak | QUASipeak | SAMPlE | AVGLog }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:DETection?
```

## Arguments

`AVERAge` displays the average data value for each pixel.

`POSitive` displays the maximum data value for each pixel. `NEGative` displays the minimum data value for each pixel. `CAVerage` displays the CISPR average value for each pixel. `CPEak` displays the CISPR peak value for each pixel.

`QUASipeak` displays the quasi-peak value for each pixel. `SAMPlE` displays the most recent sample value for each pixel. `AVGLog` displays the average data value of logs for each pixel.

The parameter <x> = 1 to 4; `MARKer0` (reference marker) is invalid. The specified marker must be activated using the `CALCulate:MARKer:ADD` command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where `MEASview` keyword is optional by default y is 1

## Examples

`TRACE1 : SPECTRUM : MEASVIEW2 : DETECTION POSitive` displays the maximum data value for each pixel on Trace 1

## TRACe<x>:SPECTrum[:MEASview<y>]:FREeze

Determines whether or not to freeze the display of the specified trace in the Spectrum view.

The parameter <x> = 1 to 5; All traces are valid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:FREeze?
```

## Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating the display of the specified trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
TRACE1 : SPECTRUM:MEASVIEW2 : FREEZE ON freezes Trace 1 in the Spectrum Analysis display.
```

## TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION

Sets or queries the function for the specified trace in the Spectrum view.

The parameter <x> = 1 to 3; Trace 4 (math trace) and Trace 5 (spectrogram) are invalid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION { NONE | AVERage | MAXHold | MINHold | AVGLog }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:FUNCTION?
```

## Arguments

NONE selects the normal spectrum display.

AVERage selects the Average display that indicates the average signal level at each frequency point.

MAXHold selects the Max Hold display that indicates the maximum signal level at each frequency point.

MINHold selects the Min Hold display that indicates the minimum signal level at each frequency point.

AVGLog selects the Average of logs display that indicates the average signal level of logs at each frequency point.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE1:SPECTRUM:MEASVIEW2:FUNCTION MAXHold selects Max Hold for Trace 1 in the Spectrum view.

## TRACe<x>:SPECTrum[:MEASview<y>]:LEFToperand

Sets or queries the left operand for the math trace (Trace 4) in the Spectrum view. The parameter <x> = 4; Only Trace 4 (math trace) is valid.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

TRACe<x>:SPECTrum[:MEASview<y>]:LEFToperand { TRACE1 | TRACE2 | TRACE3 }

TRACe<x>:SPECTrum[:MEASview<y>]:LEFToperand?

## Related Commands

TRACe<x>:SPECTrum[:MEASview<y>]:RIGHToperand

## Arguments

TRACE1 selects Trace 1 as the left operand for the math trace.

TRACE2 selects Trace 2 as the left operand for the math trace.

TRACE3 selects Trace 3 as the left operand for the math trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

TRACE4:SPECTRUM:MEASVIEW2:LEFTOPERAND TRACE1 selects Trace 1 as the left operand for the math trace.

## TRACe<x>:SPECTrum[:MEASview<y>]:RIGHToperand

Sets or queries the right operand for the math trace (Trace 4) in the Spectrum view. The parameter <x> = 4; Only Trace 4 (math trace) is valid.



## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:RIGHToperand { TRACE1 | TRACE2 | TRACE3 }
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:RIGHToperand?
```

## Related Commands

```
TRACe<x>:SPECTrum[:MEASview<y>]:LEFToperand
```

## Arguments

TRACE1 selects Trace 1 as the right operand for the math trace.

TRACE2 selects Trace 2 as the right operand for the math trace.

TRACE3 selects Trace 3 as the right operand for the math trace.

The parameter <x> = 1 to 4; MARKer0 (reference marker) is invalid. The specified marker must be activated using the CALCulate:MARKer:ADD command

The parameter <y>=1 to 8 represents measurement index based on number of sources of connected instrument, where MEASview keyword is optional by default y is 1

## Examples

```
TRACE4 : SPECTRUM:MEASVIEW2:RIGHTOPERAND TRACE1 selects Trace 1 as the right operand for the math trace
```

## TRACe<x>:SPECTrum[:MEASview<y>]:SElect

Sets or queries the trace to display the readout at the upper left of the Spectrum view.

The parameter <x> = 1 to 5; All traces are valid.



**Note:** TRACe5 (spectrogram) is valid when the spectrum and spectrogram measurements are running.

## Conditions

Measurement views: Spectrum

## Group

Trace commands

## Syntax

```
TRACe<x>:SPECTrum[:MEASview<y>]:SElect
```

```
TRACe<x>:SPECTrum[:MEASview<y>]:SElect?
```

## Arguments

None

## Returns

{ 0 | 1 }

- 0 indicates that the readout is not being displayed for the specified trace.
- 1 indicates that the readout is being displayed for the specified trace.

## Examples

TRACE1:SPECTRUM:MEASview1:SELECT selects Trace 1 to display the readout.

TRACE1:SPECTRUM:MEASview1:SELECT? might return 1, indicating that the readout is being displayed for Trace 1.

## TRACe<x>:TXGain

Sets or queries the display to show or hide the specified trace.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain { OFF | ON | 0 | 1 }
```

```
TRACe<x>:TXGain?
```

## Arguments

OFF or 0 hides the specified trace.

ON or 1 shows the specified trace.

## Returns

OFF or 0 means that the specified trace is hidden.

ON or 1 means that the specified trace is showing.

## Examples

TRACE1:TXGAIN 1 shows Trace 1.

## TRACe<x>:TXGain:AVERage:COUNT

Sets or queries the number of traces averaged to generate the displayed trace. (Present only when Function is set to anything but Normal.)

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain:AVERAge:COUNT <number>
```

## Arguments

<number>: :=<NR1> specifies the number of traces to combine for averaging.

Range: 1 to 10000

## Returns

<NR1> is an integer that is the number of traces to combine for averaging.

## Examples

```
TRACE1 : TXGAIN : AVERAGE : COUNT 64 sets the average count to 64 for Trace 1.
```

## TRACe<x>:TXGain:AVERAge:RESet (No Query Form)

Clears/resets the average data and counter and restarts the averaging process.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

Function must be is set to anything but Normal.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain:AVERAge:RESet
```

## Arguments

None.

## Examples

```
TRACE1 : TXGAIN : AVERAGE : RESET clears average data and counter, and restarts the average process for Trace 1.
```

## TRACe<x>:TXGain:COUNT

Sets or queries how many acquisitions run in the single acquisition mode for the Average or Max or Min Hold trace in the measurement.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

Function must be is set to anything but Normal.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain:COUNT <number>
```

```
TRACe<x>:TXGain:COUNT?
```

## Arguments

<number>::=<NR1> specifies the count for Averaged, Max or Min Hold.

Range: 1 to 10000.

## Returns

<NR1> is an integer that is the count for Averaged, Max or Min Hold.

## Examples

```
TRACE1 : TXGAIN : COUNT 32 sets the count to 32 for Trace 1.
```

## TRACe<x>:TXGain:COUNT:ENABLE

The command enables or disables whether to use the number of traces Averaged or Max Hold or Min Hold to generate the displayed trace. The query returns whether or not it is enabled.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain:COUNT:ENABLE { OFF | ON | 0 | 1 }
```

```
TRACe<x>:TXGain:COUNT:ENABLE?
```

## Arguments

OFF or 0 disables the count for the specified trace.

ON or 1 enables the count for the specified trace.

## Returns

OFF or 0 means that the count for the specified trace is disabled.

ON or 1 means that the count for the specified trace is enabled.

## Examples

TRACE1:TXGAIN:COUNT:ENABLE ON enables the count for Trace 1.

## TRACe<x>:TXGain:COUNT:RESet (No Query Form)

Restarts the trace count when the function is Average, Max Hold, or Min Hold.

### Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:TXGain:COUNT:RESet
```

### Arguments

None.

## Examples

TRACE1:TXGAIN:COUNT:RESET clears the trace data and counter, and restarts the process for Trace 1.

## TRACe<x>:TXGain:FREeze

Sets the trace as frozen or live. The query returns whether the trace is frozen or live.

### Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

### Group

Trace commands

### Syntax

```
TRACe<x>:TXGain:FREeze { OFF | ON | 0 | 1 }
```

```
TRACe<x>:TXGain:FREeze?
```

### Arguments

OFF or 0 updates the display of the specified trace normally.

ON or 1 stops updating (freezes) the display of the specified trace.

## Returns

OFF or 0 means that the specified trace is updating normally on the display.

ON or 1 means that the specified trace is not updating on the display.

## Examples

`TRACE1:TXGAIN:COUNT:FREEZE ON` freezes Trace 1 in the display.

## TRACe<x>:TXGain:FUNctIon

Sets or queries the trace processing method.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

## Group

Trace commands

## Syntax

`TRACe<x>:TXGain:FUNctIon { NORMal | MAXHold | MINHold | AVERage | AVGLog }`

## Arguments

`NORMal` selects the normal display.

`AVERage` selects the Average display that indicates the average signal level at each frequency point.

`MAXHold` selects the Max Hold display that indicates the maximum signal level at each frequency point.

`MINHold` selects the Min Hold display that indicates the minimum signal level at each frequency point.

`AVGLog` selects the Average of logs display that indicates the average signal level of logs at each frequency point.

## Returns

See Arguments.

## Examples

`TRACE1:TXGAIN:COUNT:FUNCTION MAXHold` selects Max Hold for Trace 1.

## TRACe<x>:TXGain:SElect

Selects or queries the specified trace.

## Conditions

Measurement view: Transmission Gain Where <x> is the trace and is 1, 2, or 3.

## Group

Trace commands

## Syntax

```
TRACe<x>:TXGain:SElect
```

```
TRACe<x>:TXGain:SElect?
```

## Arguments

None.

## Returns

The currently selected trace number.

## Examples

```
TRACE1 : TXGAIN : SELECT selects Trace 2.
```

```
TRACE1 : TXGAIN : SELECT? might return 1, indicating that Trace 1 is selected.
```

## TRACe<x>:WLAN:AD:CONStE:RESUlt:CONTent:STATe

Sets or queries the state of the specified Result Contents (Preamble, Header, Data, Guard) of the specified trace.

## Conditions

Measurement view: 802.11ad Constellation Where the <x> trace variable is 1, 2, 3, or 4

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:AD:CONStE:RESUlt:CONTent:STATe <boolean>,<arg>
```

```
TRACe<x>:WLAN:AD:CONStE:RESUlt:CONTent:STATe?
```

## Arguments

### <boolean> arguments.

0 or OFF disables the specified trace. 1 or ON enables the specified trace.

### <arg> result contents arguments.

The Result Contents arguments are:

- 1 is the Preamble
- 2 is the Header
- 3 is the Data
- 4 is the Guard (only applies to Single Carrier standard)

## Returns

See Arguments

## Examples

`TRACe1:WLAN:AD:CONStE:RESUlT:CONTEnt:STATe 1, 1` enables the Preamble result contents for trace 1.

`TRACe2:WLAN:AD:CONStE:RESUlT:CONTEnt:STATe?` might return 0, 3, indicating the Data result contents for trace 2 are disabled.

## TRACe<x>:WLAN:AY:CONStE:RESUlT:CONTEnt:STATe

Set or query the result contents to Preamble, Header, Data, and Guard for the 11ay Single Carrier standard.

### Conditions

Measurement view: 802.11ay Analysis Constellation

### Group

Trace commands

### Syntax

`TRACe<x>:WLAN:AY:CONStE:RESUlT:CONTEnt:STATe { 0 | 1 }`

where <x> is 1,2,3,4

`TRACe<x>:WLAN:AY:CONStE:RESUlT:CONTEnt:STATe?`

Where <x> is 1,2,3,4

### Arguments

OFF or 0 disables the Preamble, Header, Data, and Guard result contents are disabled for the 11ay Single Carrier standard (for 4.32 GHz BW Header is not applicable) based on Trace <x> selection, where <x> is 1,2,3,4.

ON or 1 enables the Preamble, Header, Data, and Guard result contents are enabled for the 11ay Single Carrier standard (for 4.32 GHz BW Header is not applicable) based on Trace <x> selection, where <x> is 1,2,3,4.

### Returns

OFF or 0 means that the Preamble, Header, Data, and Guard result contents are disabled for the 11ay Single Carrier standard (for 4.32 GHz BW Header is not applicable) based on Trace <x> selection, where <x> is 1,2,3,4.

ON or 1 means that the Preamble, Header, Data, and Guard result contents are enabled for the 11ay Single Carrier (for 4.32 GHz BW Header is not applicable) based on Trace <x> selection, where <x> is 1,2,3,4.

## Examples

`TRAC1:WLAN:AY:CONS:RESU:CONT:STAT ON` enables the Preamble, Header, Data, and Guard result contents for the 11ay Single Carrier standard under the Trace 1 tab.

`TRAC2:WLAN:AY:CONS:RESU:CONT:STAT?` may return OFF, which means that the Preamble, Header, Data, and Guard result contents are disabled for the 11ay Single Carrier standard under the Trace 2 tab.

## TRACe<x>:WLAN:CONStE

Turns the selected trace on or off in the constellation plot display, or queries the enabled status. <x> is accepted by the command but ignored as there is only a single WLAN constellation trace.



## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:CONStE { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:CONStE?
```

## Arguments

OFF or 0 turns off the constellation trace.

ON or 1 turns on the constellation trace

## Returns

Returns the status of the selected trace as an integral value.

0 implies a disabled trace.

1 implies a enabled trace.

## Examples

TRAC : WLAN : CONS 1 turns the constellation trace on in the plot display.

## TRACe<x>:WLAN:CONStE:FREeze

When the freeze state is turned on, the selected trace stops updating in the constellation plot. <x> is accepted by the command but ignored as there is only a single WLAN constellation trace.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:CONStE:FREeze { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:CONStE:FREeze?
```

## Arguments

ON or 1 turns on the freeze state, causing the constellation trace to stop updating.

OFF or 0 turns off the freeze state.

## Returns

Outputs the frozen status of the selected trace as an integral value. 1 indicates a frozen trace (stopped updating) and 0 a non-frozen trace.

## Examples

`TRAC : WLAN : CONS : FRE 1` freezes the constellation trace, causing it to stop updating.

## TRACe<x>:WLAN:CRESpOse:SElect

Sets or queries which WLAN Channel Response trace is the active trace. TRACe1 is Magnitude, TRACe2 is Phase.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

`TRACe<x>:WLAN:CRESpOse:SElect`

`TRACe<x>:WLAN:CRESpOse:SElect?`

## Arguments

None

## Returns

Returns 1, which implies that the selected trace is the active trace.

Returns 0, which implies that the selected trace is not the active trace.

## Examples

`TRACE1 : WLAN : CRES : SELECT` specifies to select the magnitude trace.

`TRACE1 : WLAN : CRES : SELECT?` might return 1, indicating that the magnitude trace is the active trace.

## TRACe<x>:WLAN:EVM:FREEze

When the freeze state is turned on, the selected trace stops updating. TRACe1 is Averages. TRACe2 is Points.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:EVM:FREeze {0|1|OFF|ON}
```

```
TRACe<x>:WLAN:EVM:FREeze?
```

## Arguments

ON or 1 turns on the freeze state, causing the selected trace to stop updating.

OFF or 0 turns off the freeze state.

## Returns

Outputs the frozen status of the selected trace as an integral value. 1 indicates a frozen trace and 0 a non-frozen trace.

## Examples

```
TRACE2:WLAN:EVM:FREEZE 1 causes Trace 2 to stop updating.
```

```
TRACE2:WLAN:EVM:FREEZE 1? might return 0, indicating that Trace 2 is not in a freeze state and is updating.
```

## TRACe<x>:WLAN:EVM[:SHOW]

Displays or hides the selected trace. TRACe1 is Averages. TRACe2 is Points.



**Note:** Use this command only with multicarrier standards and not singlecarrier (802.11b).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:EVM[:SHOW] {0|1|OFF|ON}
```

```
TRACe<x>:WLAN:EVM[:SHOW]?
```

## Arguments

ON or 1 displays the trace.

OFF or 0 hides the trace.

## Examples

```
TRAC2:WLAN:EVM:SHOW 1 displays the WLAN EVM points trace.
```

## TRACe<x>:WLAN:MERRor:FREeze

When the freeze state is turned on, the selected trace stops updating. TRACe1 is Averages. TRACe2 is Points.

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:MERRor:FReeze { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:MERRor:FReeze?
```

## Arguments

ON or 1 turns on the freeze state, causing the selected trace to stop updating.

OFF or 0 turns off the freeze state.

## Returns

Outputs the frozen status of the selected trace as an integral value. 1 indicates a frozen trace and 0 a non-frozen trace.

## Examples

TRACE2:WLAN:MERR:FREEZE 1 causes Trace 2 to stop updating.

TRACE2:WLAN:MERR:FREEZE 1? might return 0, indicating that Trace 2 is not in a freeze state and is updating.

## TRACe<x>:WLAN:MERRor[:SHOW]

Displays or hides the selected trace. TRACe1 is Averages. TRACe2 is Points.



**Note:** Use this command only with multicarrier standards and not singlecarrier (802.11b).

## Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:MERRor[:SHOW] { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:MERRor[:SHOW]?
```

## Arguments

ON or 1 displays the trace.

OFF or 0 hides the trace.

## Examples

TRAC2:WLAN:MERR:SHOW 1 displays the WLAN Magnitude Error points trace.

## TRACe<x>:WLAN:PERRor:FREEze

When the freeze state is turned on, the selected trace stops updating. TRACe1 is Averages. TRACe2 is Points.

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

### Group

Trace commands

### Syntax

```
TRACe<x>:WLAN:PERRor:FREEze { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:PERRor:FREEze?
```

### Arguments

ON or 1 turns on the freeze state, causing the selected trace to stop updating.

OFF or 0 turns off the freeze state.

### Returns

Outputs the frozen status of the selected trace as an integral value. 1 indicates a frozen trace and 0 indicates a non-frozen trace.

### Examples

TRACE2:WLAN:PERR:FREEZE 1 causes Trace 2 to stop updating.

TRACE2:WLAN:PERR:FREEZE 1? might return 0, indicating that Trace 2 is not in a freeze state and is updating.

## TRACe<x>:WLAN:PERRor[:SHOW]

Displays or hides the selected trace. TRACe1 is Averages. TRACe2 is Points.



**Note:** Use this command only with multicarrier standards and not singlecarrier (802.11b).

### Conditions

Measurement view: WLAN

This command requires WLAN Measurements

## Group

Trace commands

## Syntax

```
TRACe<x>:WLAN:PERRor[:SHOW] { 0 | 1 | OFF | ON }
```

```
TRACe<x>:WLAN:PERRor[:SHOW] ?
```

## Arguments

ON or 1 displays the trace.

OFF or 0 hides the trace.

## Examples

TRAC2:WLAN:PERR:SHOW 1 displays the WLAN Phase Error points trace.

# Trigger commands

## \*TRG (No Query Form)

Generates a trigger. It produces the same effect as the Force Trigger button on the Trigger control panel. This command is valid when the trigger mode is Triggered.

## Conditions

Measurement views: All

## Group

IEEE common commands

## Syntax

```
*TRG
```

## Arguments

None

## Examples

\*TRG generates a trigger.

## TRIGger:ADVanced:SWEep:MODE

Sets or enables the analyzer to wait for a trigger for each acquisition in a swept spectrum or other measurement requiring multiple acquisitions worth of span.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Trigger commands

## Syntax

```
TRIGger:ADVanced:SWEEP:MODE {OFF|ON|0|1}
```

## Arguments

ON or 1 enables the wait for a trigger for each acquisition.

OFF or 0 disables the wait for a trigger for each acquisition.

## Returns

## Examples

TRIGGER:ADVANCED:SWEEP:MODE 1 causes the analyzer to wait for a trigger for each acquisition.

## TRIGger:DPX:SHOW:FRAMES

Determines when to show frames in the swept acquisition mode. Show frames continuously when ON or show only triggered frames when OFF.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger:DPX:SHOW:FRAMES { OFF | ON | 0 | 1 }
```

```
TRIGger:DPX:SHOW:FRAMES?
```

## Arguments

OFF or 0 shows only triggered frames.

ON or 1 shows frames continuously.

## Examples

TRIGger:DPX:SHOW:FRAMES ON shows frames continuously.

## TRIGger:DPX:TDM:SHOW:ALL

Sets or queries the selections of the DPX Prefs control panel tab page.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger:DPX:TDM:SHOW:ALL { OFF | ON | 0 | 1 }
```

```
TRIGger:DPX:TDM:SHOW:ALL?
```

## Arguments

OFF or 0 selects the Synch with acquisition triggers option.

ON or 1 selects the Trigger as fast as possible option.

## Examples

TRIGGER:DPX:TDM:SHOW:ALL ON selects the Trigger as fast as possible option in the DPX Prefs control panel tab.

## TRIGger:EVENT:EXtErnal:SLOPe

Sets or queries the signal slope of the External input to determine a trigger event.

## Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.

## Group

Trigger commands

## Syntax

```
TRIGger:EVENT:EXtErnal:SLOPe { RISE | FALL }
```

## Arguments

RISE Trigger event occurs on the rising edge.

FALL Trigger event occurs on the falling edge.

## Examples

TRIGGER:EVENT:EXTERNAL:SLOPE RISE sets the trigger event to occur on the rising edge of the External signal input.

## TRIGger:EVENT:INPut:LEVel

Sets or queries the power level that the signal must pass through to cause a trigger event when the trigger type is set to Power.

## Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.



## Group

Trigger commands

## Syntax

```
TRIGger:EVENT:INPut:LEVel <numeric>
```

## Arguments

<numeric> ::= <NR3>

Range: -170 dBm to 50 dBm

## Examples

TRIGGER:EVENT:INPUT:LEVEL 10 sets the trigger threshold level to 10 dBm.

## TRIGger:EVENT:INPut:SLOPe

Sets or queries the signal slope of the RF Input to determine a trigger event.

## Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.

## Group

Trigger commands

## Syntax

```
TRIGger:EVENT:INPut:SLOPe { RISE | FALL }
```

## Arguments

RISE Trigger event occurs on the rising edge.

FALL Trigger event occurs on the falling edge.

## Examples

TRIGGER:EVENT:INPUT:SLOPE RISE sets the trigger event to occur on the rising edge of the RF Input signal.

## TRIGger:EVENT:INPut:TYPE

Sets or queries the type of trigger. (Power is the only available setting.)

## Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.

## Group

Trigger commands

## Syntax

```
TRIGger:EVENT:INPut:TYPE {POWer}
```

## Arguments

POWer Sets trigger type to Power.

## Examples

```
TRIGGER:EVENT:INPUT:TYPE POWER sets the trigger type to Power.
```

## TRIGger:EVENT:SOURce

Sets or queries the source of the signal to be monitored for a trigger event.

## Conditions

Requires an RSA 306, 306B, 500A series, or 600A series be connected.

## Group

Trigger commands

## Syntax

```
TRIGger:EVENT:SOURce { INPut | EXTernal }
```

```
TRIGger:EVENT:SOURce?
```

## Arguments

INPut Sets the trigger source to the RF Input.

EXTernal Sets the trigger source to use the External input.

## Examples

```
TRIGGER:EVENT:SOURCE INPUT sets the trigger source to the RF Input.
```

## TRIGger:MASK:NEW (No Query Form)

Loads a new frequency mask.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger:MASK:NEW <freq(1)>,<ampl(1)>,<freq(2)>,<ampl(2)>,...,<freq(n)>,<ampl(n)>(n  
= 500 maximum)
```

## Arguments

$\langle \text{freq}(n) \rangle$ ,  $\langle \text{ampl}(n) \rangle$  are numbers that represent a frequency (Hz) and amplitude (dBm) pair that define a point of the mask. Up to 500 pairs can be specified with zero frequency being the center screen. The mask is visible in the spectrum view with the following trigger conditions



**Note:** Do not include a unit suffix (such as Hz, dBm, or dB) with the arguments for this command.

- Trigger mode: Triggered (TRIGger[:SEquence]:TIME:QUALified:TIME<x> is set to ON or 1.)
- Trigger type: Frequency Mask (TRIGger[:SEquence]:EVENT:INPut:TYPE is set to FMASK.)

## Examples

TRIGGER:MASK:NEW:-8E6,-80,0,-10,8E6,-80 loads the mask with the points A (-8 MHz, -80 dBm), B (0 Hz, -10 dBm), and C (8 MHz, -80 dBm), as shown in the following figure.

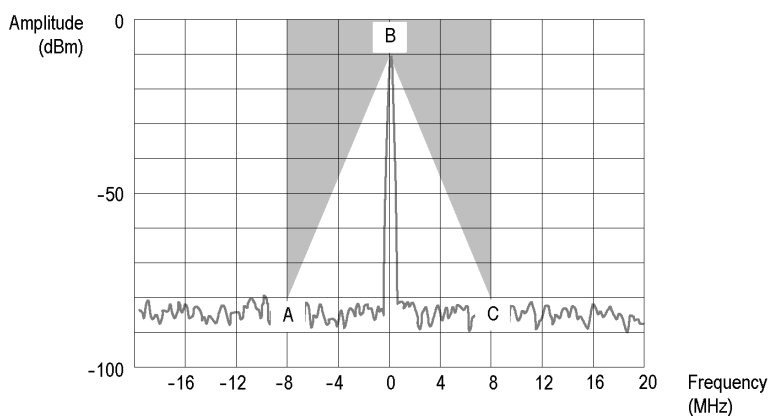


Figure 9: Trigger mask setting example

## TRIGger:MASK:NEW:AUTO (No Query Form)

Draws a new frequency mask automatically based on a reference trace.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

TRIGger:MASK:NEW:AUTO <meas\_ID>,<trace\_ID>,<x\_margin>,<y\_margin>

### Arguments

$\langle \text{meas\_ID} \rangle :: \langle \text{string} \rangle$  specifies the measurement view.

$\langle \text{trace\_ID} \rangle$  specifies the reference trace.

The values of  $\langle \text{meas\_ID} \rangle$  and  $\langle \text{trace\_ID} \rangle$  are listed in the following table for each possible measurement view.

`<x_margin>::<Nrf>` specifies the X margin (horizontal offset from the reference trace) in Hz.

`<y_margin>::<Nrf>` specifies the Y margin (vertical offset from the reference trace) in dB.

Measurement view	<meas_ID>	<trace_ID>
Spectrum	"specan"	TRACE1 (Trace 1), TRACE2 (Trace 2), TRACE3 (Trace 3), TRACE4 (Math trace), TRACE5 (Spectrogram trace)
DPX, Spectrum	"dpx"	TRACE1 (+Peak trace), TRACE2 (-Peak trace), TRACE3 (Avg trace), TRACE4 (Math trace)q
GFFFGChannel power and ACPR	"acpr"	TRACE1 (Trace 1)
MCPR	"mcpr"	TRACE1 (Trace 1)
OBW	Q"obw"	TRACE1 (Trace 1)

## Examples

`TRIGGER:MASK:NEW:AUTO"specan", TRACE 1,2E+6,15` draws a new frequency mask automatically in the Spectrum view based on Trace 1 with the horizontal margin of 2 MHz and the vertical margin of 15 dB.

## TRIGger:MASK:OPEN (No Query Form)

Opens a trigger mask with a specified file. To save a trigger mask, use the `TRIGger:MASK:SAVE` command.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

`TRIGger:MASK:OPEN <file_name>`

### Arguments

`<file_name>::<string>` specifies the trigger mask file to open.

The file extension is `.msk`. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-63.)

## Examples

`TRIGGER:MASK:OPEN"C:\My Documents\Mask1"` opens the trigger mask with the *Mask1* file in the *My Documents* directory.

## TRIGger:MASK:SAVE (No Query Form)

Saves the current trigger mask to a specified file. To open the trigger mask, use the TRIGger:MASK:OPEN command.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger:MASK:SAVE <file_name>
```

### Arguments

<file\_name>::<string> specifies the file to save the trigger mask.

The file extension is .msk. You can omit the extension.

For the directory of file, refer to *Specifying the File* (See page 2-63.)

### Examples

```
TRIGGER:MASK:SAVE"C:\My Documents\Mask1" saves the trigger mask to the Mask1 file in the My Documents directory.
```

## TRIGger:SAVE:COUNT:MAXimum

Sets or queries the number of times a Save on Trigger action stores a file. After this limit is reached, no more files are saved. The instrument will continue to run, but no additional files are saved when a trigger event occurs.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Trigger commands

### Syntax

```
TRIGger:SAVE:COUNT:MAXimum <numeric>
```

### Arguments

<numeric>::=<NR1>

Range: 1 to 10000

### Examples

```
TRIGGER:SAVE:COUNT:MAXIMUM 100 sets the number of files saved to 100.
```

## **TRIGger:SAVE:COUNT[:STATe]**

Sets or queries the Maximum total number of files feature. When enabled and the set limit is reached, no more files are saved. The instrument will continue to run, but no additional files are saved when a trigger event occurs.

### **Conditions**

Requires an RSA be connected with SignalVu-PC.

### **Group**

Trigger commands

### **Syntax**

```
TRIGger:SAVE:COUNT[:STATe] { OFF | ON | 0 | 1 }
```

### **Arguments**

ON or 1 enables the maximum number of saved files feature.

OFF or 0 disables the maximum number of saved files feature.

### **Examples**

```
TRIGGER:SAVE:COUNT[:STATE] 1 enables the maximum number of saved files feature.
```

## **TRIGger:SAVE:DATA:FORMat**

Sets or queries the file format used when saving the acquisition data on a trigger event.

### **Conditions**

Requires an RSA be connected with SignalVu-PC.

### **Group**

Trigger commands

### **Syntax**

```
TRIGger:SAVE:DATA:FORMat { TIQ | CSV | MAT }
```

### **Arguments**

TIQ Native format data file which may include multiple frames.

CSV Comma-separated values format.

MAT: MATLAB format.

### **Examples**

```
TRIGGER:SAVE:DATA:FORMAT TIQ saves the data file in native format.
```

## TRIGger:SAVE:DATA[:STATe]

Sets or queries to save data when a trigger event occurs.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Trigger commands

### Syntax

```
TRIGger:SAVE:DATA[:STATe] { OFF | ON | 0 | 1 }
```

### Arguments

ON or 1 enables the save on trigger function.

OFF or 0 disables the save on trigger function

### Examples

```
TRIGGER:SAVE:DATA[:STATE] 1 enables the save on trigger function.
```

## TRIGger:SAVE:PICTure:FORMat

Sets or queries the file format used when saving a picture of the display when a trigger event occurs.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Trigger commands

### Syntax

```
TRIGger:SAVE:PICTure:FORMat { PNG | JPG | BMP }
```

### Arguments

PNG, JPG, BMP: picture formats available.

### Examples

```
TRIGGER:SAVE:PICTURE:FORMAT PNG sets the picture format extension to .PNG.
```

## TRIGger:SAVE:PICTure[:STATe]

Sets or queries the save picture on trigger setting. This enables saving a picture of the display when a trigger event occurs.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Trigger commands

## Syntax

```
TRIGger:SAVE:PICTure[:STATe] { OFF | ON | 0 | 1 }
```

## Arguments

OFF or 0 enables saving a picture.

ON or 1 disables saving a picture.

## Examples

TRIGGER:SAVE:PICTURE[:STATE] 1 enables saving a picture of the display when a trigger event occurs.

## TRIGger[:SEQUence]:ADVanced:HOLDoff

Sets or queries the trigger holdoff time which prevents triggers until there have been no trigger events for at least the specified holdoff period.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEQUence]:ADVanced:HOLDoff <value>
```

```
TRIGger[:SEQUence]:ADVanced:HOLDoff?
```

## Arguments

<value>::<NRf> specifies the trigger holdoff time. Range: 20 ns to 10 s.

## Examples

TRIGger[:SEQUence]:ADVanced:HOLDoff 1.5us sets the trigger holdoff time to 1.5  $\mu$ s.

## TRIGger[:SEQUence]:ADVanced:HOLDoff:ENABLE

Enables or queries the status of the Advanced Trigger Holdoff function.

## Conditions

Measurement views: All



## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:ADVanced:HOLDoff:ENABle { OFF | ON | 0 | 1 }
```

```
TRIGger[:SEquence]:ADVanced:HOLDoff:ENABle?
```

## Arguments

OFF or 0 does not trigger based on the advanced trigger holdoff settings.

ON or 1 enables triggering based on the advanced trigger holdoff settings.

## Examples

TRIGger:ADVanced:HOLDoff:ENABle ON enables triggering based on the advanced trigger holdoff settings.

## TRIGger[:SEquence]:ADVanced:SWEep:MODE

Determines whether or not to trigger each segment in the swept acquisition mode.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:ADVanced:SWEep:MODE { OFF | ON | 0 | 1 }
```

```
TRIGger[:SEquence]:ADVanced:SWEep:MODE?
```

## Arguments

OFF or 0 does not trigger each segment in the swept acquisition mode.

ON or 1 triggers each segment in the swept acquisition mode.

## Examples

TRIGGER:SEQUENCE:ADVANCED:SWEEP:MODE ON triggers each segment in the swept acquisition mode.

## TRIGger[:SEquence]:ADVanced:SWEPT:SEGMENT:ENABLE

Determines whether or not to set the spectrum analyzer to wait for a trigger for each acquisition in a swept spectrum or other measurement requiring multiple acquisitions worth of span.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:ADVanced:SWEpt:SEGment:ENABle { OFF | ON| 0 | 1 }
```

```
TRIGger[:SEquence]:ADVanced:SWEpt:SEGment:ENABle?
```

## Arguments

OFF or 0 does not trigger each segment in the swept acquisition mode.

ON or 1 triggers each segment in the swept acquisition mode.

## Examples

`TRIGger[:SEquence]:ADVanced:SWEpt:SEGment:ENABle ON` waits for a trigger for each acquisition in a swept spectrum or other measurement requiring multiple acquisitions worth of span.

## TRIGger[:SEquence]:EVENT:A:EDGE:COUPling

Sets or queries the choice of trigger coupling type.

## Conditions

Measurement views: All

## Arguments

For MSO/LPD, available arguments are {DC | HFREject | LFREject | NOISereject}.

For SX, available arguments are {DC | AC | HFREject | LFREject | NOISereject| RF}.

## Group

Trigger commands

## Syntax

```
:TRIGger[:SEquence]:EVENT:A:EDGE:COUPling {DC| AC | HFREject | LFREject | NOISereject| RF}
```

```
TRIGger[:SEquence]:EVENT:A:EDGE:COUPling?
```

## Examples

`TRIGger:SEquence:EVENT:A:EDGE:COUPling DC` sets the Trigger coupling type to DC .

## TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:MODE

Sets or queries the trigger hold off mode.

### Conditions

Measurement views: All

This cmd is applicable only for SX.

### Group

Trigger commands

### Syntax

```
:TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:MODE {TIME | RANDom | AUTO}
```

```
:TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:MODE?
```

### Examples

```
:TRIGger:SEQuence:EVENT:A:EDGE:HOLDoff:MODE RANDom sets Holdoff mode to Random.
```

## TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:TIME

Sets or queries the time after triggering on an event, that the instrument waits before detecting the same trigger event to start the next acquisition.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:TIME <value>
```

```
TRIGger[:SEQuence]:EVENT:A:EDGE:HOLDoff:TIME?
```

### Range

{0 to 10} in seconds

### Examples

```
TRIGger:SEQuence:EVENT:A:EDGE:HOLDoff:TIME 5 sets the hold off time to 5 seconds.
```

## TRIGger[:SEQuence]:EVENT:A:EDGE:LEVEL

Sets or queries where the trigger point occurs for trigger source type.

## Conditions

Measurement views: All

This cmd is applicable only for SX.

## Group

Trigger commands

## Syntax

```
:TRIGger[:SEquence]:EVENT:A:EDGE:LEVEL <value>
```

```
:TRIGger[:SEquence]:EVENT:A:EDGE:LEVEL
```

## Examples

```
:TRIGger:SEquence:EVENT:A:EDGE:LEVEL -0.316 sets the trigger point to -0.316.
```

## TRIGger[:SEquence]:EVENT:EXTFront:IMPedance

Sets or queries the impedance of the external trigger input on the front panel.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:EXTFront:IMPedance { OHM50 | OHM5K }
```

```
TRIGger[:SEquence]:EVENT:EXTFront:IMPedance?
```

## Arguments

OHM50 selects 50 $\Omega$  impedance.

OHM5K selects 5 k $\Omega$  impedance.

## Examples

```
TRIGGER:SEQUENCE:EVENT:EXTFRONT:IMPEDANCE OHM 50 selects 50  $\Omega$  impedance for the external trigger input on the front panel.
```

## TRIGger[:SEquence]:EVENT:EXTFront:LEVEL

Sets or queries the trigger level at the external trigger input on the front panel.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:EXTFront:LEVel <value>
```

```
TRIGger[:SEquence]:EVENT:EXTFront:LEVel?
```

## Arguments

<value> : <NRf> specifies the trigger level. Range: -2.5 to +2.5 V.

## Examples

TRIGGER:SEQUENCE:EVENT:EXTFRONT:LEVEL 1.5 sets the trigger level to 1.5 V at the external trigger input on the front panel.

## TRIGger[:SEquence]:EVENT:EXTFront:SLOPe

Sets or queries the trigger slope of the external trigger input on the front panel.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:EXTFront:SLOPe { RISE | FALL }
```

```
TRIGger[:SEquence]:EVENT:EXTFront:SLOPe?
```

## Arguments

RISe causes the trigger event on the rising edge.

FALL causes the trigger event on the falling edge.

## Examples

TRIGGER:SEQUENCE:EVENT:EXTFRONT:SLOPE RISE causes the trigger event on the rising edge of the external input signal.

## TRIGger[:SEquence]:EVENT:EXTRear:SLOPe

Sets or queries the trigger slope of the external trigger input on the rear panel. The trigger level is fixed to the TTL threshold.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEQuence]:EVENT:EXTREar:SLOPe { RISE | FALL }
```

```
TRIGger[:SEQuence]:EVENT:EXTREar:SLOPe?
```

## Arguments

RISe causes the trigger event on the rising edge.

FALL causes the trigger event on the falling edge.

## Examples

```
TRIGGER:SEQUENCE:EVENT:EXTREAR:SLOPE RISE causes the trigger event on the rising edge of the external input signal.
```

## TRIGger[:SEQuence]:EVENT:GATed

Sets or queries the positive or negative logic for the gated trigger input on the rear panel.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEQuence]:EVENT:GATed { HIGH | LOW }
```

```
TRIGger[:SEQuence]:EVENT:GATed?
```

## Arguments

HIGH specifies that the gated trigger input is high active.

LOW specifies that the gated trigger input is low active.

## Examples

```
TRIGGER:SEQUENCE:EVENT:GATED HIGH specifies that the gated trigger input is high active.
```

## TRIGger[:SEQuence]:EVENT:INPut:DDENsity:AMPLitude

Sets or queries the center vertical level for the density trigger region of the RF input DPX bitmap display.

## Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude <value>
```

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude?
```

## Related Commands

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance
```

## Arguments

<value>::<NRf> specifies the vertical center of the trigger region in dBm.

Range: 0 to 200 dBm.

## Examples

TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude 100dBm sets the center level to 100 dBm for the density trigger region.

## TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance

Sets or queries the positive and negative extensions of the density trigger region from the center vertical level. The density trigger region is defined for the DPX bitmap display.

## Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance <value>
```

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance?
```

## Related Commands

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance
```

## Arguments

<value>::<NRf> specifies an absolute amplitude value for the positive and negative limits of the density trigger region in dBm.

Range: within the defined measurement box.

## Examples

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:AMPLitude:TOLerance 50` sets the positive limit 50 dBm above the amplitude level and the negative limit 50 dBm below the amplitude level.

## TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency

Sets or queries the center horizontal level for the density trigger region of the RF input DPX bitmap display.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency <value>`

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency?`

### Arguments

`<value>::<Nrf>` specifies the horizontal center of the trigger region. Range: within the defined measurement box.

## Examples

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency 1.82GHz` sets the center horizontal level to 1.82 GHz for the density trigger region.

## TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance

Sets or queries the left and right extensions of the density trigger region from the center horizontal level. The density trigger region is defined for the DPX bitmap display.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance <value>`

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance?`



## Arguments

`<value> : : <Nrf>` specifies an absolute frequency value for the left and right limits from the center frequency of the density trigger region. Range: within the defined measurement box.

## Examples

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:FREQuency:TOLerance 0.2GHz`, for a center frequency of 1.82 GHz, sets the left horizontal limit to 1.62 GHz and the right horizontal limit to 2.02 GHz for the density trigger region.

## TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold

Sets or queries the DPX Density threshold that defines a trigger event for the density trigger region of the DPX bitmap display. The DPX Density must be higher or lower (specified by the Density selection) than this value to define a trigger event.

## Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

## Group

Trigger commands

## Syntax

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold <value>`

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold?`

## Related Commands

## Arguments

`<value> : : <Nrf>` specifies the DPX Density threshold as a percentage (%).

Range: within the defined measurement box.

## Examples

`TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold 2` sets the DPX Density threshold to 2 % for the density trigger.

## TRIGger[:SEquence]:EVENT:INPut:DDENsity:VIOLation

Sets or queries whether a density value higher or lower than the THReshold value defines a trigger event for the density trigger region of the DPX bitmap display.

## Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:VIOLation { HIGHer| LOWer }
```

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:VIOLation?
```

## Related Commands

```
TRIGger[:SEquence]:EVENT:INPut:DDENsity:THReshold
```

## Arguments

HIGHer a DPX density value above the THReshold value defines the trigger event.

LOWer a DPX Density value below the THReshold value defines the trigger event.

## Examples

TRIGger:EVENT:INPut:DDENsity:VIOLation LOWER sets the DPX Density violation to trigger when the probability of occurrence is lower than the threshold percentage.

## TRIGger[:SEquence]:EVENT:INPut:FEDGE:LEVel

Sets or queries the trigger level at the frequency edge input.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:FEDGE:LEVel <value>
```

```
TRIGger[:SEquence]:EVENT:INPut:FEDGE:LEVel?
```

## Arguments

<value>::<NRf> specifies the trigger level.

Range:  $\pm$  (Max AcqBW/2) Hz

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:FEDGE:LEVEL 1.5 sets the trigger level to 1.5 V at the frequency edge input.

## TRIGger[:SEquence]:EVENT:INPut:FEDGE:LEVel:ACTual? (Query Only)

Queries the actual length of the frequency edge input for the DPX measurement.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:FEDGE:LEVel:ACTual?
```

## Arguments

None

## Returns

<NRf> The actual length in Hz.

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:FEDGE:LEVEL:ACTUAL? might return 10 Hz, indicating that the actual length is 10 Hz.

## TRIGger[:SEquence]:EVENT:INPut:FEDGE:SLOPe

Sets or queries the trigger slope of the frequency edge input.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:INPut:FEDGE:SLOPe { RISE | FALL }
```

```
TRIGger[:SEquence]:EVENT:INPut:FEDGE:SLOPe?
```

## Arguments

RISE causes the trigger event on the rising edge.

FALL causes the trigger event on the falling edge.

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:FEDGE:SLOPE RISE causes the trigger event on the rising edge of the frequency edge input.

## **TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:LEVel**

Set or queries the threshold level in the DPX measurement.

### **Conditions**

Measurement views: All

### **Group**

Trigger commands

### **Syntax**

```
TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:LEVel <value>
```

```
TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:LEVel?
```

### **Arguments**

<value>::<NRf> specifies the threshold level.

Range: 50 dBm to -170 dBm

### **Examples**

TRIGGER:SEQUENCE:EVENT:INPUT:FEDGE:THRESHOLD:LEVEL -25 sets the threshold level to -25 dBm.

## **TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:STATe**

Set or queries the threshold state in the DPX measurement.

### **Conditions**

Measurement views: All

### **Group**

Trigger commands

### **Syntax**

```
TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:STATe { OFF | ON | 0 | 1 }
```

```
TRIGger[:SEQuence]:EVENT:INPut:FEDGE:THReshold:STATe?
```

### **Arguments**

OFF or 0 disables the threshold.

ON or 1 enables the threshold.

### **Examples**

TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH:STATE OFF enables the threshold.

## TRIGger[:SEQuence]:EVENT:INPut:FMASk:{BANDwidth|BWIDth}[:RESolution]

Sets or queries the Resolution Bandwidth value to be used in the spectrum view for the frequency mask trigger.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:{BANDwidth|BWIDth}[:RESolution] <value>
```

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:{BANDwidth|BWIDth}[:RESolution]?
```

### Arguments

<value>::<Nrf> specifies the resolution bandwidth for Frequency Mask triggering.

### Examples

TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth 300kHz sets the resolution bandwidth to 300 kHz.

## TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth|BWIDth[:RESolution]:ACTual? (Query Only)

Queries current setting for the Resolution Bandwidth value used in the spectrum view for the frequency mask trigger.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth|BWIDth[:RESolution]:ACTual?
```

### Arguments

None.

### Examples

TRIGger:EVENT:INPut:FMASk:BWIDth:ACTUAL? returns the resolution bandwidth value.

## TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth|BWIDth[:RESolution]:AUTO

Sets or queries whether to automatically set the Resolution Bandwidth value used in the spectrum view for the frequency mask trigger.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth|BWIDth[:RESolution]:AUTO { OFF | ON | 0 | 1 }
```

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:BANDwidth|BWIDth[:RESolution]:AUTO?
```

### Arguments

OFF or 0 sets the resolution bandwidth manually with the command `TRIGger[:SEQuence]:EVENT:INPut:FMASk:{BANDwidth|BWIDth}[:RESolution]`.

ON or 1 sets the resolution bandwidth automatically.

### Examples

`TRIGger:EVENT:INPut:FMASk:BWIDth:AUTO ON` sets the resolution bandwidth automatically.

## TRIGger[:SEQuence]:EVENT:INPut:FMASk:VIOLation

Sets or queries when the analyzer triggers in the frequency mask trigger.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:VIOLation { T | F | TF | FT | TFT | FTF }
```

```
TRIGger[:SEQuence]:EVENT:INPut:FMASk:VIOLation?
```

### Arguments

The following table lists the arguments which represent the trigger violations.

**Table 55: Trigger violations**

Violation	Description
T	Only one state is required to initiate a trigger event. The signal has at least one data point inside the mask. The trigger event occurs at the first point that appears inside the mask. A trigger event could occur on the first acquisition.
F	Only one state change is required to initiate a trigger event. The signal has at least one data point outside the mask. The trigger event occurs at the first point that appears outside the mask. A trigger event could occur on the first acquisition.
TF	Two states are required to initiate a trigger event. The signal must be inside the mask and then passes out of the mask. The trigger event occurs at the first transition where the signal passes out of the mask.
FT	Two states are required to initiate a trigger event. The signal must be outside the mask and then passes into the mask. The trigger event occurs at the first transition where the signal passes into the mask.
TFT	Three states are required to initiate a trigger event. The signal starts inside the mask and then passes out of the mask. Next, the signal must pass into the mask. The trigger event occurs at the second transition where the signal passes back into the mask.
FTF	Three states are required to initiate a trigger event. The signal starts outside the mask and then passes into the mask. Next, the signal must pass back outside the mask. The trigger event occurs at the second transition where the signal passes back out of the mask.

## Examples

`TRIGGER:SEQUENCE:EVENT:INPUT:FMASK:VIOLATION TF` specifies that the analyzer will trigger when the signal has crossed into the mask and then outside of the mask.

## TRIGger[:SEQuence]:EVENT:INPut:LEVel

Sets or queries the trigger level for the RF input level trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

`TRIGger[:SEQuence]:EVENT:INPut:LEVel <value>`

`TRIGger[:SEQuence]:EVENT:INPut:LEVel?`

### Arguments

`<value>::<NRf>` specifies the trigger level. Range: -170 to +50 dBm.

### Examples

`TRIGGER:SEQUENCE:EVENT:INPUT:LEVEL -10` sets the trigger level to -10 dBm for the RF input level trigger.

## TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe

Sets or queries the Runt trigger for a positive or negative going pulse.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe { POSitive | NEGative }
```

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe?
```

### Arguments

POSitive causes the trigger event on a positive runt pulse.

NEGative causes the trigger event on a negative runt pulse.

### Examples

```
TRIGger:EVENT:INPut:RUNT:PULSE POSitive causes the trigger event on a positive pulse.
```

## TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULse:HIGH:LEVel

Sets or queries the upper level (first threshold) to qualify a Runt trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULse:HIGH:LEVel <value>
```

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULse:HIGH:LEVel?
```

### Arguments

<value> : : <NRf> sets the upper vertical level for a runt pulse in dBm.

### Examples

```
TRIGger:EVENT:INPut:RUNT:PULSE 300 causes the trigger event when the leading edge of a pulse crosses the 300 dBm level.
```



## TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe:LOW:LEVel

Sets or queries the lower level (second threshold) to qualify a Runt trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe:LOW:LEVel <value>
```

```
TRIGger[:SEQuence]:EVENT:INPut:RUNT:PULSe:LOW:LEVel?
```

### Arguments

<value>::<NRf> sets the lower vertical level for a runt pulse in dBm.

### Examples

TRIGger:EVENT:INPut:RUNT:PULSE 3 causes the trigger event when the following edge of a pulse crosses the 3 dBm level.

## TRIGger[:SEQuence]:EVENT:INPut:SLOPe

Sets or queries the trigger slope for the RF input level trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:SLOPe { RISE | FALL }
```

```
TRIGger[:SEQuence]:EVENT:INPut:SLOPe?
```

### Arguments

RISe causes the trigger event on the rising edge.

FALL causes the trigger event on the falling edge.

### Examples

TRIGGER:SEQUENCE:EVENT:INPUT:SLOPE RISE causes the trigger event on the rising edge of the RF input signal.

## TRIGger[:SEQuence]:EVENT:INPut:TDBWidth

Sets or queries the time-domain bandwidth for the RF input power trigger. This command is effective when TRIGger[:SEQuence]:EVENT:INPut:TDBWidth: STATE is ON.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth <value>
```

```
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth?
```

### Arguments

<value>::<NRf> specifies the time-domain bandwidth.

Range: 1 Hz to 60 MHz.

### Examples

TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH 5MHz sets the time-domain bandwidth to 5 MHz for the RF input power trigger.

## TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:ACTual? (Query Only)

Queries the actual time-domain bandwidth for the RF input power trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:ACTual?
```

### Related Commands

```
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth
```

### Arguments

None

### Returns

<NRf> Actual time-domain bandwidth.

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH:ACTUAL? might return  
1.000E+6, indicating that the actual time-domain bandwidth is 1 MHz.

## TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:STATe

Determines whether to set the time-domain bandwidth automatically or manually for the RF input power trigger.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:STATe { OFF | ON | 0 | 1 }
TRIGger[:SEQuence]:EVENT:INPut:TDBWidth:STATe?
```

### Arguments

OFF or 0 sets the time-domain bandwidth automatically.

ON or 1 sets the time-domain bandwidth manually using the TRIGger[:SEQuence]:EVENT:INPut:TDBWidth command.

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:TDBWIDTH:STATE OFF sets the time-domain bandwidth automatically.

## TRIGger[:SEQuence]:EVENT:INPut:TYPE

Sets or queries the trigger type for the source of RF input.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:EVENT:INPut:TYPE { POWer | FMASK | DDENsity | RUNT | FEDGe }
TRIGger[:SEQuence]:EVENT:INPut:TYPE?
```

### Arguments

POWer uses the power level for triggering.

FMASK uses the frequency mask for triggering. DDENSITY uses the density mask for triggering. RUNT uses the runt mask for triggering.

## Examples

TRIGGER:SEQUENCE:EVENT:INPUT:TYPE FMASK uses the frequency mask for triggering.

## TRIGger[:SEQuence]:EVENT:SOURce

Sets or queries the trigger event source.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

TRIGger[:SEQuence]:EVENT:SOURce { INPut | EXTFront | EXTRear | EXTGated | LINE }

TRIGger[:SEQuence]:EVENT:SOURce?

### Arguments

The following table lists the arguments.

Table 56: Trigger event source

Argument	Source
INPut	RF input
EXTFront	Trigger in (front)
EXTRear	Trigger in (rear)
EXTGated	Gate in
LINE	AC line

## Examples

TRIGGER:SEQUENCE:EVENT:SOURCE INPut specifies the trigger event source as the RF input.

## TRIGger[:SEQuence]:FORCed

Determines whether or not to cause a manual trigger if the acquisition is armed, ready and waiting for a trigger. This command is valid when

TRIGger[:SEQuence]:TIME:QUALified:TIME<x> is set to On (the trigger mode is Triggered).

### Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:FORCed { OFF | ON | 0 | 1 }
```

```
TRIGger[:SEquence]:FORCed?
```

## Arguments

OFF or 0 does not cause a manual trigger.

ON or 1 causes a manual trigger.

## Examples

TRIGGER:SEQUENCE:FORCED ON causes a manual trigger if the acquisition is armed, ready and waiting for a trigger.

## TRIGger:SEquence:FORCed (No Query Form)

Causes a trigger event to occur when the trigger mode is set to Triggered, regardless of the set trigger conditions.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Trigger commands

## Syntax

```
TRIGger:SEquence:FORCed
```

## Arguments

None

## Examples

TRIGGER:SEQUENCE:FORCED forces a trigger event.

## TRIGger[:SEquence]:IMMediate (No Query Form)

Causes a trigger immediately, skipping the event detection and delay. This command is valid when TRIGger[:SEquence]:TIME:QUALified:TIME<x> is set to On (the trigger mode is Triggered).

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

TRIGger[:SEQuence]:IMMediate

## Arguments

None

## Examples

TRIGGER:SEQUENCE:IMMEDIATE causes a trigger immediately, skipping the event detection and delay.

## TRIGger[:SEQuence]:STATus

Sets or queries the trigger mode (Free Run or Triggered).

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

TRIGger[:SEQuence]:STATus { OFF | ON | 0 | 1 }

TRIGger[:SEQuence]:STATus?

## Arguments

OFF or 0 selects the free-run mode.

ON or 1 selects the triggered mode.

## Examples

TRIGGER:SEQUENCE:STATUS ON selects the triggered mode.

## TRIGger[:SEQuence][[:STATus]

Sets or queries the trigger mode setting.

## Conditions

Requires an RSA be connected with SignalVu-PC.

## Group

Trigger commands

## Syntax

TRIGger[:SEQuence][[:STATus] { 0 | 1 }

TRIGger[:SEQuence][[:STATus]?

## Arguments

0 sets the trigger mode to Free Run.

1 sets the trigger mode to Triggered.

## Examples

TRIGGER[:SEQUENCE][:STATUS] 0 sets the trigger mode to Free Run.

## TRIGger[:SEQuence]:TIME:DELAy

Sets or queries the trigger delay time which occurs after recognizing an event but before actually declaring the trigger.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

TRIGger[:SEQuence]:TIME:DELAy <value>

TRIGger[:SEQuence]:TIME:DELAy?

## Arguments

<value>::<NRf> specifies the trigger delay time. Range: 0 to 60 s.

## Examples

TRIGGER:SEQUENCE:TIME:DELAY 1.5 sets the trigger delay time to 1.5 s.

## TRIGger[:SEQuence]:TIME:POSition

Sets or queries the position of the trigger event within the acquisition record.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

TRIGger[:SEQuence]:TIME:POSition <value>

TRIGger[:SEQuence]:TIME:POSition?

## Arguments

<value>::<NRf> specifies the trigger position. Range: 1% to 99%.

## Examples

TRIGGER:SEQUENCE:TIME:POSITION 20 sets the trigger position to 20% of the acquisition record.

## TRIGger[:Sequence]:TIME:POSition:AUTO

Sets or queries the position of the trigger event (automatic or manual) within the acquisition record.

### Conditions

Measurement views: All

### Group

Trigger commands

### Syntax

```
TRIGger[:Sequence]:TIME:POSition:AUTO { OFF | ON | 0 | 1 }
```

```
TRIGger[:Sequence]:TIME:POSition:AUTO?
```

### Arguments

OFF or 0 sets the position of the trigger event manually.

ON or 1 sets the position of the trigger event automatically.

## Examples

TRIGGER:SEQUENCE:TIME:POSITION:AUTO ON sets the position of the trigger event automatically.

## TRIGger[:SEQuence]:TIME:QUALified

Sets or queries the timing qualification setting for triggers. Trigger events that do not meet the timing qualifications are ignored.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

```
TRIGger[:SEQuence]:TIME:QUALified { SHORTer | LONGer | INSide | OUTSide | NONE }
```

```
TRIGger[:SEQuence]:TIME:QUALified?
```

### Arguments

SHORTer the length of the trigger event is shorter than the time specified by the TRIGger:TIME:QUALified:TIME1 setting.

LONGer the length of the trigger event is longer than the time specified by the TRIGger:TIME:QUALified:TIME1 setting.



INSide the length of the trigger event is longer than the time `TRIGger:TIME:QUALified:TIME1` setting and shorter than the `TRIGger:TIME:QUALified:TIME2` setting.

OUTSide the length of the trigger event is shorter than the time `TRIGger:TIME:QUALified:TIME1` setting or longer than the `TRIGger:QUALified:TIME2` setting.

## Examples

`TRIGger:SEQuence:TIME:QUALified SHORTer` the length of the trigger event is shorter than the time specified by the `TRIGger:TIME:DELAY` setting.

## TRIGger[:SEQuence]:TIME:QUALified:TIME<x>

Sets or queries the Time1 and Time2 parameters for time qualified triggering. Time1 is the shorter or first time, and Time2 is the second or later time for the qualifying period. Each `TIME<x>` must be set with a separate command.

### Conditions

Measurement views: All

For SignalVu-PC users, the RSA instrument must be connected.

### Group

Trigger commands

### Syntax

`TRIGger[:SEQuence]:TIME:QUALified:TIME<x> <value>`

### Related Commands

`TRIGger[:SEQuence]:TIME:QUALified`

### Arguments

`<value>::<NRf>` the length of the trigger time for TIME1 or TIME2. (Range for RSA7100A with SignalVu-PC: 0 ns to 1 s.)

(Range for RSA5100B: 0 ns to 28 s.)

### Examples

`TRIGger:TIME:QUALified:TIME1 55µs` sets the TIME1 trigger point at 55 µs after the trigger event.

## TRIGger[:SEQuence]:EVENT:A:EDGE:SETTings

Sets or queries the setting of trigger source type.

### Conditions

Measurement views: All

This cmd is applicable only for SX.

## Group

Trigger commands

## Syntax

```
:TRIGger[:SEquence]:EVENT:A:EDGE:SOURce:TYPE {SHARed|INDEpendent}  
:TRIGger[:SEquence]:EVENT:A:EDGE:SOURce:TYPE?
```

## Arguments

Shared specifies that all trigger settings will be shared among all trigger sources.

Independent specifies that all trigger settings will be independent of trigger source.

## Examples

```
:TRIGger:SEquence:EVENT:A:EDGE:SOURce SHARed sets the source type settings to shared.
```

## TRIGger[:SEquence]:EVENT:A:EDGE:SLOPe

Sets or queries whether instrument finds the trigger point on rising or falling edge or either edge of a signal.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence]:EVENT:A:EDGE:SLOPe {RISE | FALL | EITHer}  
TRIGger[:SEquence]:EVENT:A:EDGE:SLOPe?
```

## Examples

```
TRIGger:SEquence:EVENT:A:EDGE:SLOPe RISE sets the trigger point on Rise.
```

## TRIGger[:SEquence]:EVENT:A:EDGE:SOURce

Sets or queries the channel on which signal is to be monitored for a trigger event.

## Conditions

Measurement views: All

For MSO/LPD trigger source can be selected from the list of available channels (CH1 to CH8).

For SX trigger source can be selected from list of channels as well as AUX and LINE inputs if available.

## Group

Trigger commands

## Syntax

```
:TRIGger[:SEquence]:EVENT:A:EDGE:SOURce {CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8 | MCH1 | MCH2 | MCH3 | MCH4 | AUX | LINE}
```

```
:TRIGger[:SEquence]:EVENT:A:EDGE:SOURce?
```

## Examples

TRIGger:SEquence:EVENT:A:EDGE:SOURce CH1 sets CH1 on which trigger event to be monitored.

## TRIGger[:SEquence]:EVENT:A:TYPE

Sets or queries the trigger type { supporting only Edge trigger}.

## Conditions

Measurement views: All

## Group

Trigger commands

## Syntax

```
TRIGger[:SEquence:]EVENT:A:TYPE { EDGE }
```

```
TRIGger[:SEquence:]EVENT:A:TYPE?
```

## Examples

TRIGger:SEquence:EVENT:A:TYPE EDGE sets trigger type to edge.

## TRIGger[:SEquence]:MODE

Sets or queries the trigger mode setting.

## Conditions

This cmd is applicable only for SX.

## Group

Trigger commands

## Syntax

```
:TRIGger[:SEquence]:MODE {0 | 1}
```

```
:TRIGger[:SEquence]:MODE?
```

## Examples

`:TRIGger:SEQuence:MODE 0` sets the trigger mode to Free Run.

## TRIGger:TIME:POSition

Sets or queries the point (in percent) along the acquisition record the trigger event should occur. Manually setting the trigger position disables the Auto setting.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Trigger commands

### Syntax

`TRIGger:TIME:POSition <numeric>`

### Arguments

`<numeric>::=<NR1>`

Range: 1 to 100

## Examples

`TRIGGER:TIME:POSITION 50` sets the acquisition point to 50%.

## TRIGger:TIME:POSition:AUTO

Sets or queries the instrument to automatically set the position of the trigger event.

### Conditions

Requires an RSA be connected with SignalVu-PC.

### Group

Trigger commands

### Syntax

`TRIGger:TIME:POSition:AUTO { OFF | ON | 0 | 1 }`

### Arguments

ON or 1 enables auto trigger position.

OFF or 0 disables auto trigger position.

## Examples

TRIGGER:TIME:POSITION:AUTO 1 enables the auto trigger position setting.

## UNIT:POWer

Sets or queries the fundamental unit of power. This command is equivalent to [SENSE]:POWer:UNITs.

### Conditions

Measurement views: All

### Group

Unit commands

### Syntax

UNIT:POWer { DBM | DBV | VOLTs | WATTs | DBUW | DBW | DBUV | DBMV | DBUA | DBUV\_M | DBUA\_M | AMPS }UNIT:POWer?

### Arguments

The following table lists the arguments.

**Table 57: Table 2-48: Power units**

Argument	Power unit
DBM	dBm
DBV	dBV
VOLTs	Volts
WATTs	Watts
DBUW	dB $\mu$ W
DBW	dBW
DBUV	dB $\mu$ V
DBMV	dBmV
DBUA	dB $\mu$ A
DBUV_M	dB $\mu$ V/m
DBUA_M	dB $\mu$ A/m
AMPS	Amps



**Note:** Select dB $\mu$ V/m or dB $\mu$ A/m unit when using an antenna table.

## Examples

UNIT:POWERDBM specifies the fundamental unit of power as dBm.

## **\*WAI (No Query Form)**

Prevents the analyzer from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the analyzer with your application program. For the details, refer to Synchronizing Execution (See page 3-11.).

### **Conditions**

Measurement views: All

### **Group**

IEEE common commands

### **Syntax**

\*WAI

### **Related Commands**

\*OPC

### **Arguments**

None

## Status and Events

The SCPI interface in the analyzer includes a status and event reporting system that enables the user to monitor crucial events that occur in the instrument. The analyzer is equipped with four registers and one queue that conform to IEEE Std 488.2-1987. This section will discuss these registers and queues along with status and event processing.

### Status and event reporting system

The following figure outlines the status and event reporting mechanism offered in the RSA Series analyzers. It contains three major blocks

- Standard Event Status
- Operation Status
- Questionable Status (fan-out structure)

The processes performed in these blocks are summarized in the Status Byte. The three blocks contain four types of registers as shown in the following table.

**Table 58: Register type**

Register	Description
Condition register	Records event occurrence in the instrument. Read only.
Transition register (positive/negative)	<p>A positive transition filter allows an event to be reported when a condition changes from false to true.</p> <p>A negative filter allows an event to be reported when a condition changes from true to false.</p> <p>Setting both positive and negative filters true allows an event to be reported anytime the condition changes.</p> <p>Clearing both filters disables event reporting.</p>
Event register	Records events filtered by the transition register. Read only.
Enable register	Masks the event register to report in the summary bit. User-definable.

\* The use of Bit 15 is not allowed in SCPI.  
The value of this bit is always zero.

CR: Condition Register  
TR: Transition Register  
EVR: Event Register  
ENR: Enable Register

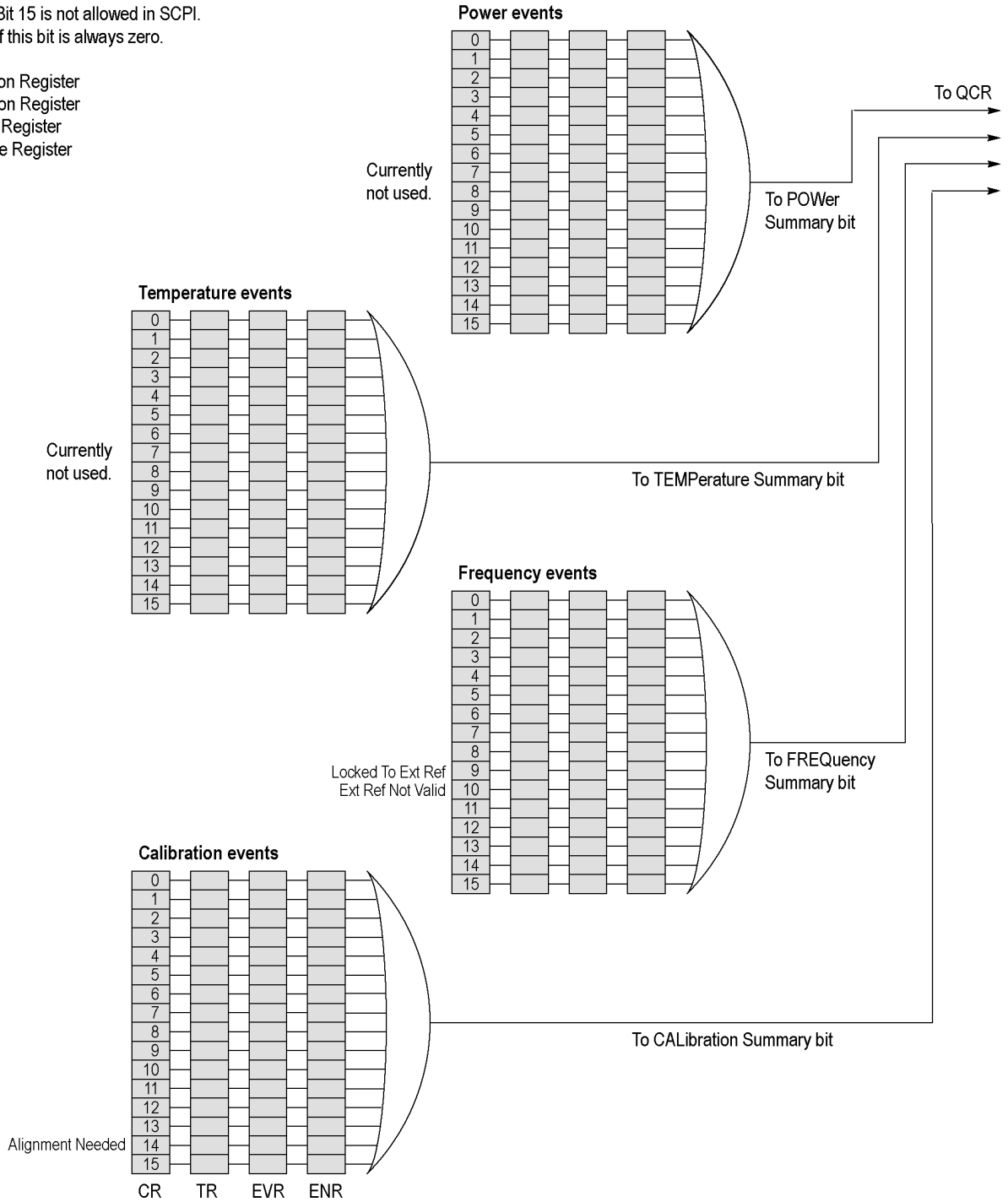


Figure 10: Status/Event reporting mechanism



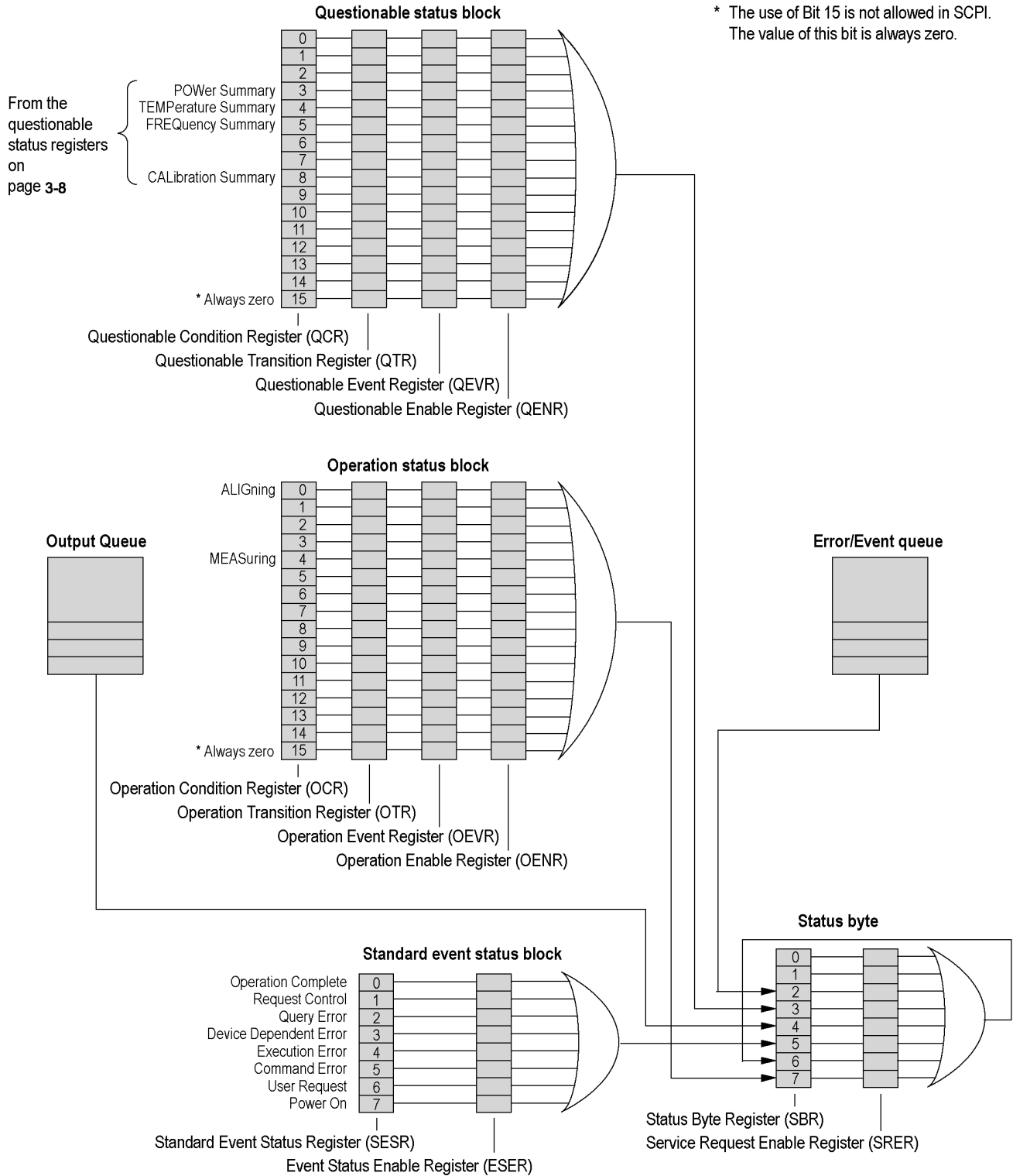


Figure 11: Status/Event reporting mechanism (Cont.)

## Status Byte

The Status Byte contains the following two registers

- Status Byte Register (SBR)
- Service Request Enable Register (SRER)

### Status Byte Register (SBR)

The SBR is made up of 8 bits. Bits 4, 5 and 6 are defined in accordance with IEEE Std 488.2-1987. These bits are used to monitor the output queue, SESR and service requests, respectively. The contents of this register are returned when the \*STB? query is used.

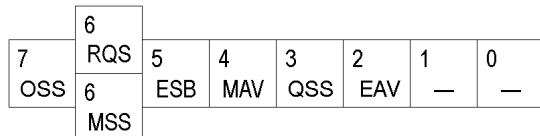


Figure 12: Status Byte Register (SBR)

Table 59: SBR bit functions

Bit	Description
7	Operation Summary Status (OSS). Summary of the operation status register.
6	Request Service (RQS)/Master Status Summary (MSS). When the instrument is accessed using the GPIB serial poll command, this bit is called the Request Service (RQS) bit and indicates to the controller that a service request has occurred (in other words, that the GPIB bus SRQ line is LOW). The RQS bit is cleared when serial poll ends.  When the instrument is accessed using the *STB? query, this bit is called the Master Status Summary (MSS) bit and indicates that the instrument has issued a service request for one or more reasons. The MSS bit is never cleared to 0 by the *STB? query.
5	Event Status Bit (ESB). This bit indicates whether or not a new event has occurred after the previous Standard Event Status Register (SESR) has been cleared or after an event readout has been performed.
4	Message Available Bit (MAV). This bit indicates that a message has been placed in the output queue and can be retrieved.
3	Questionable Summary Status (QSS). Summary of the Questionable Status Byte register.
2	Event Quantity Available (EAV). Summary of the Error Event Queue.
1-0	Not used

### Service Request Enable Register (SRER)

The SRER is made up of bits defined exactly the same as bits 0 through 7 in the SBR as shown in the following figure. This register is used by the user to determine what events will generate service requests.

The SRER bit 6 cannot be set. Also, the RQS is not maskable.

The generation of a service request with the GPIB interface involves changing the SRQ line to LOW and making a service request to the controller. The result is that a status byte for which an RQS has been set is returned in response to serial polling by the controller.

Use the \*SRE command to set the bits of the SRER. Use the \*SRE? query to read the contents of the SRER. Bit 6 must normally be set to 0.

7	6	5	4	3	2	1	0
OSB	—	ESB	MAV	QSB	—	—	—

Figure 13: Service Request Enable Register (SRER)

## Standard event status block

Reports the power on/off state, command errors, and the running state. It consists of the following registers of the following registers

- Standard Event Status Register (SESR)
- Event Status Enable Register (ESER)

These registers are made up of the same bits defined in the following figure and table. Use the \*ESR? query to read the contents of the SESR. Use the \*ESE() command to access the ESER.

7	6	5	4	3	2	1	0
PON	—	CME	EXE	DDE	QYE	—	OPC

Figure 14: Standard event status register

Table 60: Standard event status register bit definition

Bit	Description
7	Power On (PON). Indicates that the power to the instrument is on.
6	Not used.
5	Command Error (CME). Indicates that a command error has occurred while parsing by the command parser was in progress.
4	Execution Error (EXE). Indicates that an error occurred during the execution of a command. Execution errors occur for one of the following reasons <ul style="list-style-type: none"> <li>• When a value designated in the argument is outside the allowable range of the instrument, or is in conflict with the capabilities of the instrument.</li> <li>• When the command could not be executed properly because the conditions for execution differed from those essentially required.</li> </ul>
3	Device-Dependent Error (DDE). An instrument error has been detected.

Table continued...

Bit	Description
2	<p>Query Error (QYE). Indicates that a query error has been detected by the output queue controller. Query errors occur for one of the following reasons</p> <ul style="list-style-type: none"> <li>An attempt was made to retrieve messages from the output queue, despite the fact that the output queue is empty or in pending status.</li> <li>The output queue messages have been cleared despite the fact that they have not been retrieved.</li> </ul>
1	Not used.
0	Operation Complete (OPC). This bit is set with the results of the execution of the *OPC command. It indicates that all pending operations have been completed.

When an event occurs, the SESR bit corresponding to the event is set, resulting in the event being stacked in the Error/Event Queue. The SBR OAV bit is also set. If the bit corresponding to the event has also been set in the ESER, the SBR ESB bit is also set. When a message is sent to the Output Queue, the SBR MAV bit is set.

## Operation status block

The operation status block contains conditions that are part of the instrument's normal operation. It consists of the following registers

- Operation Condition Register (OCR)
- Operation Positive/ Negative Transition Register (OPTR/ONTR)
- Operation Event Register (OEVR)
- Operation Enable Register (OENR)

These registers are made up of the same bits defined in the following table and figure. Use the STATus:OPERation commands to access the operation status register set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											MEAS				ALIG

Figure 15: Operation status register

Table 61: Operation status register bit definition

Bit	Description
15	Always zero (0).
14 - 5	Not used.
4	<p>Measuring (MEAS). Indicates that the instrument is actively measuring. When the measurement ends after this bit is set in measurement, it is reset.</p> <p>"In measurement" means that one of the following commands is in execution:</p> <ul style="list-style-type: none"> <li>INITiate commands</li> <li>READ commands</li> </ul>
3 - 1	Not used.

Table continued...

Bit	Description
0	Aligning (ALIG). Indicates that the instrument is currently performing an alignment. When the alignment ends after this bit is set in alignment, it is reset.

When the specified state changes in the OCR, its bit is set or reset. This change is filtered with a transition register, and the corresponding bit of the OEVR is set. If the bit corresponding to the event has also been set in the OENR, the SBR OSS bit is also set.

## Questionable Status Block

The questionable status register set contains bits which give an indication of the quality of various aspects of the signal together with the fanned out registers as described in the next subsections. It consists of the following registers

- Questionable Condition Register (QCR)
- Questionable Positive/Negative Transition Register (QPTR/QNTR)
- Questionable Event Register (QEVR)
- Questionable Enable Register (QENR)

These registers are made up of the same bits defined in the following table and figure. Use the STATus:QUEStionable commands to access the questionable status register set.

15	14 CW	13	12	11	10	9	8 CAL	7	6	5 FREQ	4 TEMP	3 POW	2	1	0
----	----------	----	----	----	----	---	----------	---	---	-----------	-----------	----------	---	---	---

Figure 16: Questionable status register

Table 62: Questionable status register bit definition

Bit	Description
15	Always zero (0).
14	Command Warning (CW). Indicates a non-fatal warning that relates to the instrument's interpretation of a command, query, or one or more parameters of a specific command or query.
13 - 9	Not used.
8	CALibration Summary (CAL). Summary of the Questionable Calibration register.
7, 6	Not used.
5	FREQuency Summary (FREQ). Summary of the Questionable Frequency register.
4	TEMPerature Summary (TEMP). Summary of the Questionable Temperature register.
3	POWer Summary (POW). Summary of the Questionable Power register.
2 - 0	Not used.

When the specified state changes in the QCR, its bit is set or reset. This change is filtered with a transition register, and the corresponding bit of the QEVR is set. If the bit corresponding to the event has also been set in the QENR, the SBR QSS bit is also set.

### Questionable Power Register Set

Refines the power error for the POWER bit in the QCR.

*Currently not used*

### Questionable Temperature Register Set

Refines the temperature error for the TEMPerature bit in the QCR.

*Currently not used.*

### Questionable Frequency Register Set

The questionable frequency register set is made up of bits defined in the following table and figure. It refines the frequency error for the FREQUency bit in the QCR.

Use the STATUS:QUESTIONable:FREQUency commands to access the questionable frequency register set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ERNV	LTER									

Figure 17: Questionable frequency status register

**Table 63: Questionable frequency status register bit definition**

Bit	Description
15	Always zero (0).
14 - 11	Not used.
10	External Ref Not Valid (ERNV). Indicates that the external reference signal is not valid so the instrument is no longer locked to it.
9	Locked To External Ref (LTER). Indicates that the instrument is locked to the external reference signal.
8 - 0	Not used.

### Questionable Calibration Register Set

The questionable calibration register set is made up of bits defined in the following table and figure. It refines the calibration error for the CALibration bit in the QCR. Use the STATUS:QUESTIONable:CALibration commands to access the questionable calibration register set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALN														

Figure 18: Questionable calibration status register

**Table 64: Questionable calibration status register bit definition**

Bit	Description
15	Always zero.
14	Alignment Needed (ALN). Indicates the instrument needs the alignment.
13 - 0	Not used.

## Queues

There are two types of queues in the status reporting system used in the analyzer: output queues and event queues.

### Output Queue

The output queue is a FIFO (first in, first out) queue and holds response messages to queries, where they await retrieval. When there are messages in the queue, the SBR MAV bit is set.

The output queue will be emptied each time a command or query is received, so the controller must read the output queue before the next command or query is issued. If this is not done, an error will occur and the output queue will be emptied; however, the operation will proceed even if an error occurs.

### Event Queue

The event queue is a FIFO queue and stores events as they occur in the analyzer. If more than 32 events occur, event 32 will be replaced with event code -350 ("Queue Overflow"). The error code and text are retrieved using the SYSTem:ERRor queries.

## Status and event processing sequence

The following figure shows an outline of the sequence for status and event processing.

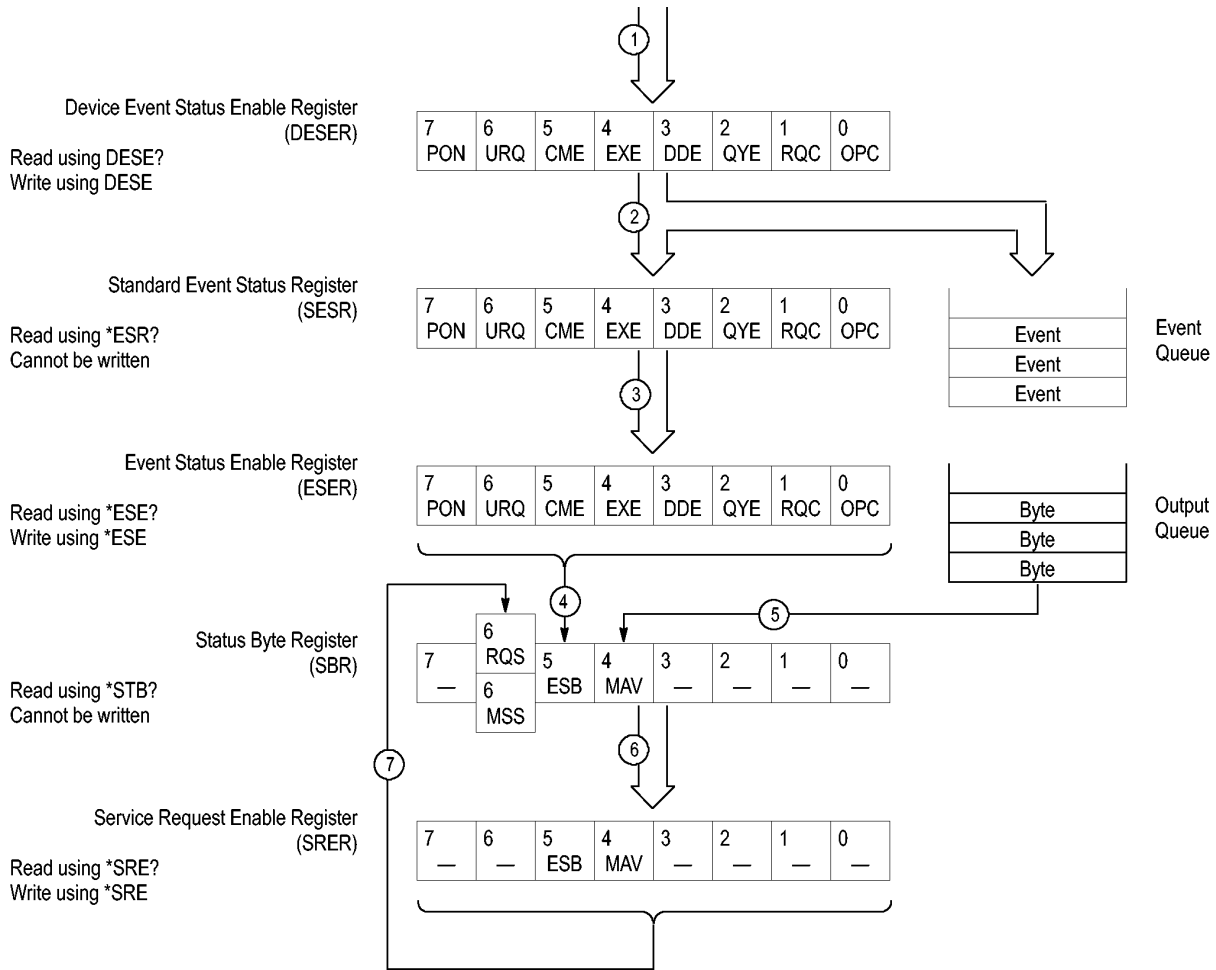


Figure 19: Status and event processing sequence

1. If an event has occurred, the SESR bit corresponding to that event is set and the event is placed in the event queue.
2. A bit corresponding to that event in the ESER has is set.
3. The SBR ESB bit is set to reflect the status of the ESER.
4. When a message is sent to the output queue, the SBR MAV bit is set.
5. Setting either the ESB or MAV bits in the SBR sets the respective bit in the SRER.
6. When the SRER bit is set, the SBR MSS bit is set and a service request is generated when using the GPIB interface.

## Synchronizing execution

Almost all commands are executed in the order in which they are sent from the controller, and the execution of each command is completed in a short period of time. However, the following commands perform data analysis in another thread, and another command can thus be executed concurrently

- INITiate commands
- READ commands
- [SENSe]:REAnalyze



These commands are designed so that the next command to be sent is executed without waiting for the previous command to be completed. In some cases, a process executed by another command must first be completed before these commands can be executed; in other cases, these commands must be completed before the next command is executed.

You have two options to achieve command synchronization

- Using the status and event reporting function
- Using synchronizing commands

## Using the Status and Event Reporting Function

In the following example, a READ command is used to obtain the measurement results while the Operation Condition Register (OCR) is being used to provide synchronization.

```

STATus:OPERation:NTRansition 16
// Set the filter of the OCR MEASuring bit
STATus:OPERation:ENABLE 16
// Enable the filter of the OCR MEASuring bit
*SRE 128
// Set the SRER OSS bit
READ:SPECTrum:TRACe
// Obtain the measurement results

```

The command waits for generation of SRQ.

## Using Synchronizing Commands

The IEEE-488.2 common commands include the following synchronizing commands

- \*OPC
- \*OPC?
- \*WAI

Using the \*OPC command. The \*OPC command sets the SESR OPC bit when all the operations for which it is waiting are completed. If the GPIB interface is in use, you can synchronize the execution by using this command together with the serial polling or service request function.

The following is a command sequence example:

```

*ESE 1
// Enable the ESER OPC bit
*SRE 32
// Enable the SRER ESB bit
ABORt;INITiate:IMMediate;*OPC
// Wait for SRQ to provide synchronization

```

Using the \*OPC? query. The query \*OPC? writes ASCII code "1" into the Output Queue when all operations for which it is waiting are completed. You can provide synchronization using the command string as the following example:

```

ABORt;INITiate:IMMediate;*OPC

```

The command waits until "1" is written into the Output Queue. When the command goes to the Output Queue to read the data, a time-out may occur before the data is written into the queue.

Using the \*WAI Command. After the process of the preceding command is completed, the \*WAI command begins to execute the process of the next command as the following example:

```
ABORt;INITiate:IMMediate;*WAI
// Wait for the *WAI process to provide synchronization
```

## Error messages and codes

Error codes with a negative value are SCPI standard error codes; errors with a positive value are unique to the RSA Series Signal Analyzers.

Event codes and messages can be obtained by using the queries `SYSTEM:ERROR?` and `SYSTEM:ERROR:ALL?` These are returned in the following format

```
<event_code>,"<event_message>"
```

## Command errors

Command errors are returned when there is a syntax error in the command.

**Table 65: Command errors**

Error code	Error message
-100	Command error
-101	Invalid character
-102	Syntax error
-103	Invalid separator
-104	Data type error
-105	GET not allowed
107	Can't replay - settings not compatible with data record
-109	Missing parameter
-111	Header separator error
-112	Program mnemonic too long
-113	Undefined header
-114	Header suffix out of range
-120	Numeric data error
-121	Character
-123	Exponent too large
-124	Too many digits
-128	Numeric data not allowed
-130	Suffix error
-131	Invalid suffix
-134	Suffix too long
-138	Suffix not allowed

Table continued...

Error code	Error message
-140	Character data error
-141	Invalid character data
-144	Character data too long
-148	Character data not allowed
-150	String data error
-151	Invalid string data
-158	String data not allowed
-160	Block data error
-161	Invalid block data
-168	Block data not allowed
-170	Command expression error
-171	Invalid expression
-178	Expression data not allowed
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error

## Execution errors

These error codes are returned when an error is detected while a command is being executed.

**Table 66: Execution errors**

Error code	Error message
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to RTL
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data

Table continued...

Error code	Error message
-224	Illegal parameter value
-225	Out of memory
-226	Lists not same length
-230	Data corrupt or stale
-231	Data questionable
-240	Hardware error
-241	Hardware missing
-250	Mass storage error
-251	Missing mass storage
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full
-256	Filename not found
-257	Filename error
-258	Media protected
-260	Execution expression error
-261	Math error in expression
-270	Execution macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label
-274	Execution macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error

## Device specific errors

These error codes are returned when an internal instrument error is detected. This type of error may indicate a hardware problem.

**Table 67: Device specific errors**

Error code	Error message
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory lost
-313	Calibration memory lost
-314	Save/Recall memory lost
-315	Configuration memory lost
-330	Self test failed
-350	Queue overflow

## Query Errors

These error codes are returned in response to an unanswered query.

**Table 68: Query errors**

Error code	Error message
-400	Query error
-410	Query interrupted
-420	Query unterminated
-430	Query deadlocked
-440	Query unterminated after indefinite period

## Device errors

These error codes are unique to the RSA Series. They are classified into three groups: global, measurement, and source conditions, as shown in the following tables.

**Table 69: Device errors, global condition**

Error code	Error message
100	Setup error
101	Disabled: data is from swept acquisition
102	Disabled: swept settings; Acquire data while display is selected
103	Acquisition bandwidth too small for current setup
104	Can't get acquisition data record

Table continued...

Error code	Error message
105	Can't open the requested display
106	Analysis failure
107	Analysis length was limited
108	Analysis length too small for current setup
109	No math trace: unmatched trace lengths
110	Analysis time was adjusted
111	Not enough samples for current setup
112	Can't replay. Data is from swept acquisition.
113	Can't replay. Live data needed for swept settings.
114	Recall error: setup not completely restored
115	Recall failure: problem with file or file contents
116	Save failure: file not saved
117	Unexpected software error. Please cycle power and try again.
118	Export failure: file not saved
119	Export failure: unable to open results file for export. File not saved.
120	Search condition for this result is already defined.
121	Search condition for this result was not found.
124	Load failed: <filename>
125	Store error: file not saved.
126	No Math trace: unmatched trace X range
127	Not enough memory for measurement
128	Incomplete analysis
129	Not enough samples for current setup
130	Mask creation error: <reason message>

**Table 70: Device errors, measurement condition**

Error code	Error message
1000	TDBW actual (TDBW: Time Domain Bandwidth)
1002	RBW increased to
1003	RBW limited by acquisition bandwidth to
1004	RBW conflict. Increase span or analysis length
1011	Audio disabled: configuration problem
1012	Audio Demod disabled: swept acquisition
1013	Audio Demod disabled: trigger in use
1014	Audio disabled: IF band outside Acq BW
1015	Calibration error. See Windows Event Viewer for error detail.
OBW errors	
Table continued...	

Error code	Error message
1016	Analysis failure: AcqBW < MeasBW + (5 x RBW)
1017	Analysis failure: AcqBW must be 10 kHz or more
1018	x dB BW > Meas BW
Pulse errors	
1019	AcqBW too low for current Chirp BW setting
1020	Impulse Response only valid for Chirp modulation
1023	Not enough memory for measurement
Other measurements	
1024	BW actual (limited by Acq BW)
1025	CISPR not available in FastFrame. Uncheck FastFrame in the Acquire panel.
1026	Analysis length must be in auto.
1027	Carrier not found
1029	CISPR accuracy limited by acq memory. Adjust RBW or freq range.
1030	CISPR: Acq BW too small for RBW. Try increasing span or freq range.
1031	Insufficient data for CISPR. Acquire while display is selected.
1032	VBW increased - Analysis Length too short
1033	VBW does not use full Analysis Length.
1034	Analysis time was adjusted Reached sample limit of

Table 71: Device errors, source condition

Error code	Error message
2008	Not calibrated
2009	FMT RBW limited by Acq BW to <XX>
2010	FMT RBW increased by Acq BW to <XX>
2011	Hardware failure - see Windows Event Viewer
2012	FMT disabled: Acq BW too small for current setup
2013	Hardware failure detected by diagnostics
2014	FMT disabled: RefLev too low for Volts/Watts units
2015	FMT disabled: settings conflict with selected measurement
2016	Preamp freq range exceeded
2024	RBW limited by Span to <XX>
2025	RBW increased by Span to <XX>
2028	External frequency reference signal not valid. Using internal reference.
2029	Unable to lock to external frequency reference. Using internal reference.
2033	Alignment error. Please run alignment again. If the problem persists, contact your Tektronix Service Center.
2034	Operational error: unable to complete operation. Please try again. If the problem persists, contact your Tektronix Service Center. See Windows Event Viewer for error detail.

Table continued...

Error code	Error message
2035	Hardware error: unable to configure hardware. Please try again. If the problem persists, contact your Tektronix Service Center. See Windows Event Viewer for error detail.
2036	Shutting down - internal temperature is too high. Check fans and airflow. If the problem persists, contact your Tektronix Service Center.
2037	Hardware error detected. To clear error, exit and restart the application. If the problem persists, contact your Tektronix Service Center.
2038	Disabled: RefLev too low\nfor Volts/Watts units
2042	Ext Corr > 20 dB pk-pk in acq segment
2044	Combined External Correction tables exceed the 60 dB peak-to-peak limit. External Correction tables were disabled. Please check table values and try again.
2045	Disabled: FastFrame doesn't support swept settings.
2046	Attenuator use currently exceeds the cautionary limit of 1200 changes per hour. Operation was stopped to protect against premature wear-out. The monitor function can be temporarily disabled in the Amplitude control panel or over the programmatic interface. Specified lifetime for an attenuator is 10,000,000 changes.
2047	Disabled: settings conflict with selected measurement
2048	Done saving — restart with Stop, then Run. The maximum number of files saved per run have been saved. To restart another FastSave acquisition, send INITIATE.

## Status Conditions

Status conditions do not appear in the event queue, they are only returned by a `:STATus:<measurement>EVENTs? query`.

### Global status

Global status conditions can be returned with any `:STATus:<measurement>EVENTs? query`.

**Table 72: Global status conditions**

Status code	Status message
10107	Analysis Length was limited
10108	Analysis length too small for current setup
10110	Analysis time was adjusted
10122	Search result not enabled
10123	Search result no longer available

### Measurement status

Measurement status conditions are unique to specific measurement groups.



**Table 73: Measurement status conditions**

Status code	Status message
Pulse Measurements	
11001	Avg Tx only available in power units
11005	Pulse detection error
11006	Setup error: Define > Measurement Point is too late
11007	No pulses found
11008	No FFT (not all pulses have results)
11020	No FFT (not enough pulses)
11022	Analysis time was adjusted - reached time limit
11039	No sidelobe found
GP Digital Modulation	
11009	No burst detected
11034	Display not valid for selected modulation type
11035	Samples per symbol > 200. Decrease Measurement BW
11036	Too many samples per symbol. Decrease Measurement BW
Chan Power and ACPR	
11010	Measuring Noise
Other	
11028	CISPR: (progress message)
11037	Ext corrections enabled for traces
Settling Time	
11041	Signal not settled
11042	Settled duration too short
11043	First transition not found
11044	Meas Length limited by Analysis Length to xx.x xs
11045	Not enough samples: increase Meas BW or Meas Length
11046	Measurement Length exceeds limit of xxM points

## Source status

Source status conditions can be returned with any :STATUS:<measurement>EVENTS? query.

**Table 74: Source status conditions**

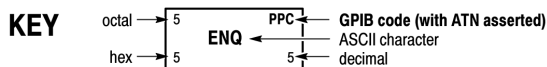
Status code	Status message
12000	Data acquired during RF ADC overrange
12005	Aligning
12006	Not aligned
Table continued...	

Status code	Status message
12007	Unaligned data
12009	Data from uncalibrated instrument
12010	Dither: manual control
12012	Data acquired during hardware failure
12014	Data acquired during RF dig gain overflow
12015	Locking to ext freq ref signal...
12018	RF Attenuator: manual control
12019	Saving acquisition
12020	Recalling acquisition data
12022	Disabled: data is from swept acquisition
12023	Disabled: Freq Mask Trigger in use
12026	Acq Sampling params: manual control
12027	Swept: RF Trig invalid for most signals
12030	Data acquired during RF ADC overrange
12031	Data acquired during RF dig gain overflow
12039	Decrease RefLev or increase\nAtten to prevent data errors
12040	Possible data errors - acquired with RefLev too much higher than RFAtten
12043	Ext Corr error - tables not applied
12056	Warming up
12057	Data acquired during warm-up
12063	Triggers disabled when DPX is swept
12064	Waiting for trigger. Open the DPX display.
12066	Data exceeds preamp range

# Appendices

## Appendix A: Character charts

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL	20 DLE	40 SP	60 0	100 @	120 P	140 `	160 p
0 0 0 1	1 GTL SOH	21 LL0 DC1	41 !	61 1	101 A	121 Q	141 a	161 q
0 0 1 0	2 STX	22 DC2	42 "	62 2	102 B	122 R	142 b	162 r
0 0 1 1	3 ETX	23 DC3	43 #	63 3	103 C	123 S	143 c	163 s
0 1 0 0	4 SDC EOT	24 DC4	44 \$	64 4	104 D	124 T	144 d	164 t
0 1 0 1	5 PPC ENQ	25 PPU NAK	45 %	65 5	105 E	125 U	145 e	165 u
0 1 1 0	6 ACK	26 SYN	46 &	66 6	106 F	126 V	146 f	166 v
0 1 1 1	7 BEL	27 ETB	47 '	67 7	107 G	127 W	147 g	167 w
1 0 0 0	10 GET BS	30 SPE CAN	50 (	70 8	110 H	130 X	150 h	170 x
1 0 0 1	11 TCT HT	31 SPD EM	51 )	71 9	111 I	131 Y	151 i	171 y
1 0 1 0	12 LF	32 SUB	52 *	72 :	112 J	132 Z	152 j	172 z
1 0 1 1	13 VT	33 ESC	53 +	73 ;	113 K	133 [	153 k	173 {
1 1 0 0	14 FF	34 FS	54 ,	74 <	114 L	134 \	154 l	174
1 1 0 1	15 CR	35 GS	55 -	75 =	115 M	135 ]	155 m	175 }
1 1 1 0	16 SO	36 RS	56 .	76 >	116 N	136 ^	156 n	176 ~
1 1 1 1	17 SI	37 US	57 /	77 ?	117 O	137 _	157 o	177 RUBOUT (DEL)
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES	TALK ADDRESSES	SECONDARY ADDRESSES OR COMMANDS			



**Tektronix**  
REF: ANSI STD X3.4-1977  
IEEE STD 488.1-1987  
ISO STD 646-2973

## Appendix B: SCPI conformance information

All commands for the RSA Series analyzers are based on SCPI Version 1999.0. The following table lists the commands that are defined in the SCPI 1999.0 Standard. The other commands not listed in the table are not defined in the SCPI 1999.0 Standard.

**Table 75: SCPI 1999.0-defined commands**

Command group	Command				
<b>IEE comman</b>	*CAL				
	*CLS				
	*ESE				
	*ESR				
	*IDN				
	*OPC				
	*OPT				
	*RST				
	*SRE				
	*STB				
	*TRG				
	*WAI				
<b>ABORt</b>	:ABORt				
<b>INITiate</b>	:INITiate	:CONTInuous			
		[:IMMediate]			
<b>STATus</b>	:STATus	:OPERation	:CONition		
			:ENABLe		
			[:EVENT]?		
			:NTRansition		
			:PTRansition		
		:PRESet			
		:QUESTionable	:CONDition?		
			:ENABLe		
			[:EVENT]?		
			:NTRansition		
			:PTRansition		
<b>SYSTem</b>	:SYSTem	:COMMunicate	:GPIB	[:SELF]	:ADDRess
		:DATE			
		:ERRor	:ALL?		
			:CODE	:ALL??	
				[:NEXT]?	
			:COUNT?		
			[:NEXT]?		
		:PREset			
:TIME					
:VERSion?					

Table continued...

Command group	Command				
:UNIT	:UNIT	:POWer			

## Appendix C: Creating a DPX Frame Access Plugin

A DPX Frame Access plugin gives access to DPX frames (bitmap and traces) from within the RSA process. The frame is accessed directly from its storage in memory, and so this method provides faster access to this data than from the ASCII (GPIB) interface.

A DPX Frame Access plugin consists of a managed assembly (dll) that contains a class that implements the IDPXFrameAccess interface. See below:

```

/// <summary>
/// If an external assembly is found that contains a type
/// that implements this interface, then it will be given
/// access to DPX frame data synchronous with data
/// coming from the hardware.
/// <summary>
public interface IDPXFrameAccess : IDisposable
{
/// <summary>
/// A frame is available. It will get called for
/// every stitch.
/// <summary>
/// <param name="frame">valid only during the duration
/// of the function call </param>
void FrameAvailable(IDPXFrame frame);
}

```

The object that is passed into the FrameAvailable function is accessed via its IDPXFrame interface. See below:

```

/// <summary>
/// A DPX frame
///
/// <summary>
public interface IDPXFrame
{
/// <summary>
/// Start frequency for traces and bitmap, Hz
/// <summary>
double StartFrequency { get; }
/// <summary>
/// Stop frequency for traces and bitmap, Hz
/// </summary>
double StopFrequency { get; }
/// <summary>
/// Top value of graph - current units
/// </summary>
double TopOfGraph { get; }
/// <summary>
/// Bottom value of graph - current units
/// </summary>
double BottomOfGraph { get; }
}

```

```

/// <summary>
/// What type of frame?
/// </summary>
DPXFrameType FrameType { get; }
/// <summary>
/// This is true if the frame is completed.
/// It will be false if swept, and not all
/// of the intermediate stitches have been
/// completed. It will be true when all stitches
/// have been completed, or if real-time.
/// </summary>
bool FrameComplete { get; }
/// <summary>
/// Get dimensions for bitmap
/// </summary>
/// <param name="rowCount"></param>
/// <param name="columnCount"></param>
void GetDimensions(out int rowCount, out int
columnCount);
/// <summary>
/// <param name="x"></param>
void GetReferenceToBitmap(out ushort[] x);
/// <summary>
/// Get reference to array holding bitmap.
/// Do not change the values in this array
/// </summary>
/// <param name="x"></param>
void GetReferenceToBitmap(out float[] x);
/// <summary>
/// Get copy of Trace1. This will be the Max trace
/// </summary>
/// <param name="yValues">function will allocate
/// the array, as necessary</param>
void GetTrace1(ref float[] yValues);
/// <summary>
/// Get copy of Trace2. This will be the Min trace
/// </summary>
/// <param name="yValues">function will allocate
/// the array, as necessary</param>
void GetTrace2(ref float[] yValues);
/// <summary>
/// Get copy of Trace3. This will be the Avg trace
/// </summary>
/// <param name="yValues">function will allocate
/// the array, as necessary</param>
void GetTrace3(ref float[] yValues);
}
public enum DPXFrameType
{
/// <summary>
Standard,
/// </summary>
Enhanced,
}

```

When the DPX measurement is running within the RSA application, the `IDPXFrameAccess.FrameAvailable` function is called once for every frame that becomes available from the DPX processing hardware. When the DPX measurement is operating in Real Time mode, `FrameAvailable` will be called for every Real Time frame. When the DPX measurement is operating in Swept mode, the total frequency span is covered by stitching together multiple real time spans at increasing center frequencies. In this case, `FrameAvailable` will be called for every stitch, with `IDPXFrame.FrameComplete` set to true on the last stitch.

The `DPXFrameAccess` assembly should not use the reference to the `IDPXFrame` object outside of the `FrameAvailable` function. It is only valid within the context of the `FrameAvailable` function call.

The creator of a `DPXFrameAccess` plugin should refer to the information in this programmer manual about the format of bitmap and trace data returned in the `IDPXFrame` object. Specifically, the `FETCH:DPX:TRACe:xxx?` queries are applicable. Traces that are not enabled are not computed and will not be available in the `IDPXFrame`.

Cautions about processing time within `FrameAvailable` function – in Real Time mode, DPX frames become available at a constant rate. As more time is taken in the `FrameAvailable` function, less CPU time is available for processing RF acquisitions (all the other measurements). If too much time is taken in the `FrameAvailable` function, frames may be missed and the update rate of other measurements may significantly slow down as they operate at lower priority than the DPX measurement.

To create a DPX Frame Access plugin, start with the example file attached to the PDF version of this manual. It has a class that implements `IDPXFrameAccess` and gives examples of accessing the various properties and methods of `IDPXFrame`.



**Note:** An example text file, `DPXFrameAccessExample.cs`, is attached to this PDF file. To access the attached file, open the PDF file and click the paperclip icon on the lower-left side of the document viewer.

Note that the build of this assembly will need to reference the `MeasurementsShared.dll` assembly in the `C:\Program Files\Tektronix\XXXX\RSA\Client` directory, where `XXXX` is either `RSA5100B`, or `SignalVu`, depending on the product..

Copy the DPX Frame Access assembly you have created to the `C:\Program Files\Tektronix\XXXX\RSA\Plugins\DPX` directory (you will need to create the directory). When the DPX measurement is started, it will discover the assembly and the `FrameAvailable` function will be called.

# Glossary

## Glossary

### **ASCII**

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

### **ASK**

Acronym for Amplitude Shift Keying. The process, or result of a process, in which the amplitude of the carrier is varied in accordance with the state of a digital input signal.

### **BNF (Backus-Naur Form)**

A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

### **Controller**

A computer or other device that sends commands to and accepts responses from the analyzer.

### **EVM (Error Vector Magnitude)**

The magnitude of an error of an actual signal relative to an ideal signal in a constellation display.

### **GPIB**

Acronym for General Purpose Interface Bus, the common name for the communications interface system defined in IEEE Std 488.

### **IEEE**

Acronym for the Institute for Electrical and Electronic Engineers.

### **IS95**

Acronym for Interim Standard-95. The standards name for first-generation CDMA cellphone technology.

### **Modulation**

The process of varying some characteristic of a signal with a second signal.

### **PSK**

Acronym for Phase Shift Keying. The process, or result of a process, in which the carrier phase is varied discretely in accordance with a digital code.

### **QAM**

Acronym for Quadrature Amplitude Modulation. The process, or result of a process, in which the amplitude and phase of the carrier are varied concurrently by synthesizing two orthogonal ASK waves (see ASK).