Tektronix®

Radar Signals Application Plug-in

Programmer Manual

077-1356-00

Tektronix®

Radar Signals Application Plug-in

Programmer Manual

www.tek.com

077-1356-00

Contacting Tektronix

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

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit www.tek.com to find contacts in your area.

Table of Contents

Getting Started

Syntax and Commands

Getting Started

Introduction

This programmer manual provides information on how to use commands for remotely controlling the Radar Signals plug-in application.

The Radar Signals plug-in can be installed in the SourceXpress software application, AWG70000 series instruments, and AWG5200 series instruments.

Communication with the plug-in is through the programmatic interface of these products. Using a single VISA or raw socket session, it is possible to communicate with SourceXpress, AWG70000 series instruments, and AWG5200 series instruments.

For information on the Remote Control, GPIB Parameters, LAN Parameters, Connecting to the Instrument using GPIB, and Setting up GPIB Communication, refer to the *AWG70000 series Arbitrary Waveform Generators Programmer Manual* or the *AWG5200 series Arbitrary Waveform Generators Programmer Manual*.

Documentation

In addition to this Radar Signals Programmer Guide, the following documentation is included with this application:

- Radar Signals Help. The help provides in-depth operation and user interface help.
- Radar Signals User Manual (PDF). This is adapted from the Radar Signals help system.

Syntax and Commands

Command Syntax

For information on the Syntax Overview, Command and Query Structure, Clearing the Instrument, Command Entry, Parameter Types, SCPI Commands and Queries, refer to the *AWG70000 series Arbitrary Waveform Generators Programmer Manual* or the *AWG5200 series Arbitrary Waveform Generators Programmer Manual*.

Command Groups

Control commands

Table 2-1: Control commands and their descriptions

Command	Description
WPLugin:ACTive	Sets or returns the active waveform plug-in.
RADar:RESet	Resets the Radar Signals plug-in by setting all the values within the module to their default values.

Antenna commands

Table 2-2: Antenna commands and their descriptions

Command	Description
RADar:PULSe:ANTenna:BEAM:AZIMuth	Sets or returns the antenna Beam Azimuth for the selected pulse definition.
RADar:PULSe:ANTenna:BEAM:ELEVation	Sets or returns the antenna Beam Elevation for the selected pulse definition.
RADar:PULSe:ANTenna:BEAM:TYPE	Sets or returns the antenna Beam Type for the selected pulse definition.
RADar:PULSe:ANTenna:SCAN:AREA	Sets or returns the antenna Scan Area for the selected pulse definition.
RADar:PULSe:ANTenna:SCAN:FILE	Sets or returns the filepath to the custom antenna scan file for the selected pulse definition.
RADar:PULSe:ANTenna:SCAN:RATE	Sets or returns the antenna Scan Rate for the selected pulse definition.
RADar:PULSe:ANTenna:SCAN:TYPE	Sets or returns the antenna scan Type (antenna beam pattern) for the selected pulse definition.
RADar:PULSe:ANTenna:TARGet:HORizontal	Sets or returns the antenna Horizontal Target position for the selected pulse definition.
RADar:PULSe:ANTenna:TARGet:VERTical	Sets or returns the antenna Vertical Target position for the selected pulse definition.
RADar:PULSe:ANTenna:TON	Sets or returns the Antenna settings state (enabled or disabled) for the selected pulse definition.

Compile commands

Table 2-3: Compile commands and their descriptions

Command	Description
RADar:COMPile	Compiles and generates waveforms and sequences using the Radar plug-in compile settings.
RADar:COMPile:CANCel	Cancels a compilation currently in progress.
RADar:COMPile:CASSign	Sets or returns the state (enabled or disabled) to compile the waveform and immediately assign it to a specified channel (enabled) or just compile the waveform (disabled).
RADar:COMPile:OSAMpling	Sets or returns the over sampling rate used to determine the sampling rate of the compiled signal.
RADar:COMPile:OVERwrite	Sets or returns the Overwrite existing waveforms/sequences state (enabled or disabled) when compiling.
RADar:COMPile:PLAY	Sets or returns the Play after assign state (enabled or disabled) for the Compile Settings.
RADar:COMPile:SEQuence	Sets or returns the state (enabled or disabled) to create sequences at end of the compile. If the sequence status is off, then each pulse train is created as one single waveform.
RADar:COMPile:SRATe	Sets or returns the manual Sampling Rate for the compile settings.
RADar:COMPile:SRATe:AUTO	Sets or returns the sampling rate Auto Calculate state (enabled or disabled).
RADar:PTRain:CHANnel:I	Sets or returns the playout channel intended for the compiled I signal waveform when the signal format is set to IQ. The selected channel is also used to define the amplitude ranges.
RADar:PTRain:CHANnel:Q	Sets or returns the playout channel intended for the compiled Q signal waveform when the signal format is set to IQ. The selected channel is also used to define the amplitude ranges.
RADar:PTRain:CHANnel:RF	Sets or returns the playout channel intended for the compiled RF waveforms and sequences when the signal format is set to RF/IF. The selected channel is also used to define the amplitude ranges.
RADar:PTRain:CORRection:APPLy	Sets or returns the Apply Corrections File state (enabled or disabled) for the Compile Settings.
RADar:PTRain:CORRection:PATH	Sets or returns the RF Correction filename and filepath to use when compiling an RF signal.
RADar:PTRain:CORRection:PATH:I	Sets or returns the I Correction filename and filepath to use when compiling and IQ signal.
OFDM:PTRain:CORRection:PATH:IQ	Sets or returns the IQ Correction filename and filepath to use when compiling and IQ signal.
RADar:PTRain:CORRection:PATH:Q	Sets or returns the Q Correction filename and filepath to use when compiling and IQ signal.
RADar:PTRain:CORRection:TYPE	Sets or returns the type of IQ correction file (a single IQ file or individual I and Q files) to apply when compiling.

Table 2-3: Compile commands and their descriptions (cont.)

Command	Description
RADar:PTRain:DUPConverter	Sets or returns the Internal IQ Modulator state (enabled or disabled) when compiling.
RADar:SFORmat	Sets or returns the signal format of the signal to be compiled.

Hopping commands

Table 2-4: Hopping commands and their descriptions

Command	Description
RADar:PULSe:HOPPing:CUSTom:RLISt	Sets or returns the Repeat List state (enabled or disabled) for the Frequency Hop table.
RADar:PULSe:HOPPing[:LIST]:ADD	Adds a single entry to the end of the Frequency Hop List or the Frequency Avoid List.
RADar:PULSe:HOPPing[:LIST]:AOFFset	Sets or returns the Amplitude Offset (in dB) of the currently selected hop.
RADar:PULSe:HOPPing[:LIST]:COUNt?	Returns the number of hops in the Hop List or the number of entries in the Frequency Avoid List.
RADar:PULSe:HOPPing[:LIST]:DELete	Removes a single entry or all entries within the Frequency Hop List or the Frequency Avoid List.
RADar:PULSe:HOPPing[:LIST]:FOFFset	Sets or returns the Frequency Offset (Relative Frequency) of the currently selected hop.
RADar:PULSe:HOPPing[:LIST]:FREQuency	Sets or returns the Frequency of the currently selected hop in the Frequency Hop List.
RADar:PULSe:HOPPing[:LIST]:INSert	Inserts a single Hop to the Frequency Hop List at the specified index.
RADar:PULSe:HOPPing[:LIST]:PSINdex	Sets or returns the Pulse Start Index (Start Index) of the currently selected hop.
RADar:PULSe:HOPPing[:LIST]:SELect	Sets or returns the selected Hop number in the Frequency Hop List or the Index number in the Frequency Avoid List.
RADar:PULSe:HOPPing:PATTern	Sets or returns the Hopping Pattern.
RADar:PULSe:HOPPing:PRBS	Sets or returns the Hopping PRBS pattern.
RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial	Sets or returns the Hopping PRBS Polynomial Expression for the User Defined PRBS.
RADar:PULSe:HOPPing:PRBS:UDEFined:RESet	Resets the Hopping User Defined PRBS Polynomial Expression and Shift register initial values to their default values.
RADar:PULSe:HOPPing:PRBS:UDEFined:SREGister	Sets or returns the Hopping PRBS Polynomial Expression for the User Defined PRBS.
RADar:PULSe:HOPPing:RANGe:ALISt[:ENABle]	Sets or returns the Hopping Frequency Avoid List state (enabled or disabled) when the Hopping Pattern is set to Pseudo List.
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MAXimum	Sets or returns the Maximum Frequency value for the Pseudo Random Range Hopping Pattern.

Table 2-4: Hopping commands and their descriptions (cont.)

Command	Description
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MINimum	Sets or returns the Minimum Frequency value for the Pseudo Random Range Hopping Pattern.
RADar:PULSe:HOPPing:RANGe[:FREQuency]:SPACing	Sets or returns the Frequency Spacing value for the Pseudo Random Range Hopping Pattern.
RADar:PULSe:HOPPing:TIME	Sets or returns the Hop Time type.
RADar:PULSe:HOPPing:TIME:PPHop	Sets or returns the Pulses per Hop for frequency hopping.
RADar:PULSe:HOPPing:TON	Sets or returns the frequency Hopping state (enabled or disabled).

IQ impairments commands

Table 2-5: IQ impairments commands and their descriptions

Command	Description
RADar:PULSe:IQIMpairment:CLEAkage:IOFFset	Sets or returns the carrier leakage I Offset percentage for IQ Impairments.
RADar:PULSe:IQIMpairment:CLEAkage:IQOFfset	Sets or returns the carrier leakage IQ Offset (in dB) for IQ Impairments.
RADar:PULSe:IQIMpairment:CLEAkage:QOFFset	Sets or returns the carrier leakage Q Offset percentage for IQ Impairments.
RADar:PULSe:IQIMpairment:CLEAkage:TON	Sets or returns the carrier leakage state (enabled or disabled) for IQ Impairments.
RADar:PULSe:IQIMpairment:IINVert	Sets or returns the I Invert state (enabled or disabled) for IQ Impairments.
RADar:PULSe:IQIMpairment:IQIMbalance:IMBalance	Sets or returns the Imbalance percentage for IQ Imbalance for IQ Impairments.
RADar:PULSe:IQIMpairment:IQIMbalance:TON	Sets or returns the IQ Imbalance state (enabled or disabled) for IQ Impairments.
RADar:PULSe:IQIMpairment:IQSWap:TON	Sets or returns the Swap I & Q state (enabled or disabled) for IQ Impairments.
RADar:PULSe:IQIMpairment:QERRor:ERRor	Sets or returns the I/Q Error percentage for Quadrature Error for IQ Impairments.
RADar:PULSe:IQIMpairment:QERRor:TON	Sets or returns the Quadrature Error state (enabled or disabled) for IQ Impairments.
RADar:PULSe:IQIMpairment:QINVert	Sets or returns the Q Invert state (enabled or disabled) for IQ Impairments.

Interference addition commands

Table 2-6: Interference addition commands and their descriptions

Command	Description
RADar:PULSe:INTerference:ANOIse:BANDwidth	Sets or returns the additive noise Bandwidth.
RADar:PULSe:INTerference:ANOIse:POTOnly	Sets or returns the additive noise Pulse On Time Only state (enabled or disabled).
RADar:PULSe:INTerference:ANOIse:SNR	Sets or returns the additive noise SNR (in dB).
RADar:PULSe:INTerference:ANOIse:TON	Sets or returns the Additive Noise interference state (enabled or disabled).

Multipath commands

Table 2-7: Multipath commands and their descriptions

Command	Description
RADar:PULSe:MULTipath:ADD	Adds the specified number of paths in the Multipath table.
RADar:PULSe:MULTipath:AMPLitude	Sets or returns the Multipath Amplitude of the currently selected path in the multipath table.
RADar:PULSe:MULTipath:DELay	Sets or returns the Multipath Delay in seconds of the currently selected path in the multipath table.
RADar:PULSe:MULTipath:PHASe	Sets or returns the Multipath Phase of the currently selected path in the multipath table.
RADar:PULSe:MULTipath:REMove	Deletes the specified index (row) in the Multipath table.
RADar:PULSe:MULTipath:SELect	Sets or returns the selected MultiPath table index (row).
RADar:PULSe:MULTipath:TON	Sets or returns the Multipath state (enabled or disabled).

Offsets commands

Table 2-8: Offsets commands and their descriptions

Command	Description
RADar:PULSe:OFFSets:AMPLitude:FILE	Sets or returns the amplitude offset User Defined Filepath.
RADar:PULSe:OFFSets:AMPLitude:IOFFset	Sets or returns the Initial Amplitude Offset value (in dB) when the amplitude offset type is set to Step Offset.
RADar:PULSe:OFFSets:AMPLitude:OFFSet	Sets or returns the Amplitude Offset value (in dB) for the Fixed amplitude offset type.
RADar:PULSe:OFFSets:AMPLitude:REPeat	Sets or returns the amplitude offset Repeat Offsets state (enabled or disabled).
RADar:PULSe:OFFSets:AMPLitude:STEP	Sets or returns the amplitude offset Step Increment value (in dB).
RADar:PULSe:OFFSets:AMPLitude:TON	Sets or returns the Amplitude Offset state (enabled or disabled).
RADar:PULSe:OFFSets:AMPLitude:TYPE	Sets or returns the amplitude offset Type.

Table 2-8: Offsets commands and their descriptions (cont.)

Command	Description
RADar:PULSe:OFFSets:FREQuency:FILE	Sets or returns the frequency offset User Defined Filepath.
RADar:PULSe:OFFSets:FREQuency:IOFFset	Sets or returns the Initial Frequency Offset value (in Hz) when the frequency offset type is set to Step Offset.
RADar:PULSe:OFFSets:FREQuency:OFFSet	Sets or returns the Frequency Offset value (in Hz) for the Fixed frequency offset type.
RADar:PULSe:OFFSets:FREQuency:REPeat	Sets or returns the frequency offset Repeat Offsets state (enabled or disabled).
RADar:PULSe:OFFSets:FREQuency:STEP	Sets or returns the frequency offset Step Increment value (in Hz).
RADar:PULSe:OFFSets:FREQuency:TON	Sets or returns the Frequency Offset state (enabled or disabled).
RADar:PULSe:OFFSets:FREQuency:TYPE	Sets or returns the frequency offset Type.
RADar:PULSe:OFFSets:PHASe:FILE	Sets or returns the phase offset User Defined Filepath.
RADar:PULSe:OFFSets:PHASe:IOFFset	Sets or returns the Initial Phase Offset value (in degrees) when the phase offset type is set to Step Offset.
RADar:PULSe:OFFSets:PHASe:OFFSet	Sets or returns the Phase Offset value (in degrees) for the Fixed phase offset type.
RADar:PULSe:OFFSets:PHASe:REPeat	Sets or returns the phase offset Repeat Offsets state (enabled or disabled).
RADar:PULSe:OFFSets:PHASe:STEP	Sets or returns the phase offset Step Increment value (in degrees).
RADar:PULSe:OFFSets:PHASe:TON	Sets or returns the Phase Offset state (enabled or disabled).
RADar:PULSe:OFFSets:PHASe:TYPE	Sets or returns the phase offset Type.

Pulse envelope commands

Table 2-9: Pulse envelope commands and their descriptions

Command	Description
RADar:PULSe:PENVelope[:CUSTom]:PATH	Sets or returns the filepath to the custom pulse shape file for the pulse envelope.
RADar:PULSe:PENVelope[:CUSTom]:SRATe	Sets or returns the pulse envelope Sampling Rate for the samples defined in the custom pulse shape file.
RADar:PULSe:PENVelope:DROOp	Sets or returns the pulse envelope deviation Droop percentage.
RADar:PULSe:PENVelope:FTIMe:PERCentage	Sets or returns the Fall Time Percentage of the pulse envelope.
RADar:PULSe:PENVelope:FTIMe:VALue	Sets or returns the Fall Time of the pulse envelope.
RADar:PULSe:PENVelope:OSHoot	Sets or returns the pulse envelope Overshoot percentage.
RADar:PULSe:PENVelope:OTIMe	Sets or returns the pulse envelope Off Time.
RADar:PULSe:PENVelope:PRF	Sets or returns the pulse envelope Pulse Repetition Frequency (PRF).
RADar:PULSe:PENVelope:PRI	This command sets or returns the pulse envelope Pulse Repetition Interval (PRI).

Table 2-9: Pulse envelope commands and their descriptions (cont.)

Command	Description
RADar:PULSe:PENVelope:RFRequency	Sets or returns the pulse envelope deviation Ripple Frequency.
RADar:PULSe:PENVelope:RIPPle	Sets or returns the pulse envelope deviation Ripple percentage.
RADar:PULSe:PENVelope:RTIMe:PERCentage	Sets or returns the Rise Time Percentage of the pulse envelope.
RADar:PULSe:PENVelope:RTIMe:VALue	Sets or returns the Rise Time of the pulse envelope.
RADar:PULSe:PENVelope:SHAPe	Sets or returns the Shape of the pulse envelope.
RADar:PULSe:PENVelope:WIDTh:PERCentage	Sets or returns the percentage at which the pulse width value is applied to the pulse shape for the pulse envelope.
RADar:PULSe:PENVelope:WIDTh:VALue	Sets or returns the Pulse Width of the pulse envelope.

Pulse impairments commands

Table 2-10: Pulse impairments commands and their descriptions

Command	Description
RADar:PULSe:IMPairments:EJITter:DEViation	Sets or returns the deviation for the Edge Jitter Pulse Impairment.
RADar:PULSe:IMPairments:EJITter:DISTribution	Sets or returns the Edge Jitter Distribution type Pulse Impairment.
RADar:PULSe:IMPairments:EJITter:TON	Sets or returns the Edge Jitter Pulse Impairments state (enabled or disabled).
RADar:PULSe:IMPairments:WJITter:DEViation	Sets or returns the Width Jitter Deviation Pulse Impairment.
RADar:PULSe:IMPairments:WJITter:DISTribution	Sets or returns the Width Jitter Distribution type Pulse Impairment.
RADar:PULSe:IMPairments:WJITter:TON	Sets or returns the Width Jitter Pulse Impairments state (enabled or disabled).

Pulse modulation commands

Table 2-11: Pulse modulation commands and their descriptions

Command	Description
RADar:PULSe:MODulation	Sets or returns the Modulation Type.
RADar:PULSe:MODulation:BARKer:CODE	Sets or returns the Barker code of the Barker modulation.
RADar:PULSe:MODulation:CSEQuence:CCOUnt	Sets or returns the Number of Chirps for the Chirp Sequence modulation.
RADar:PULSe:MODulation:CSEQuence:FSWeep	Sets or returns the Chirp Sequence modulation Frequency Sweep method.
RADar:PULSe:MODulation:CSEQuence:SRANge	Sets or returns the Chirp Sequence modulation Sweep Range.
RADar:PULSe:MODulation:CUSTom:PATH	Sets or returns the filepath to the custom modulation file.
RADar:PULSe:MODulation:DMODulation:DATA	Sets or returns the Data Pattern type for the selected Digital Modulation.

Table 2-11: Pulse modulation commands and their descriptions (cont.)

Command	Description
RADar:PULSe:MODulation:DMODulation:DATA:FILE	Sets or returns the filepath to the user provided data file for the Digital Modulation.
RADar:PULSe:MODulation:DMODulation:DATA:PATTern	Sets or returns the user defined Data Pattern for the Digital Modulation.
RADar:PULSe:MODulation:DMODulation:DATA:PRBS	Sets or returns the PRBS Data Pattern for the Digital Modulation.
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:POLYnomial	Sets or returns the PRBS Polynomial Expression for the User Defined PRBS for the Digital Modulation.
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:RESet	Resets the User Defined PRBS polynomial expression and shift register to their default values for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:SREGister	Sets or returns the PRBS Shift register initial value for the User Defined PRBS for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:FILTer:ALPHa	Sets or returns the Alpha (filter roll off) value for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern	Sets or returns the Pre/Post Pattern type of the filter for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST	Sets or returns the ending pattern (Post Pattern) of the filter for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE	Sets or returns the starting pattern (Pre Pattern) of the filter for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:FILTer:TYPE	Sets or returns the Filter Type for the selected Digital Modulation.
RADar:PULSe:MODulation:DMODulation:SSYMbols	Sets or returns the state (enabled or disabled) to use the same symbols across all repeated pulses.
RADar:PULSe:MODulation:DMODulation:SYMBols	Sets or returns the number of symbols used while applying the selected Digital Modulation on the pulse.
RADar:PULSe:MODulation:FRANk:CLENgth	Sets or returns the Frank code length when the Modulation Phase is set to Frank Code.
RADar:PULSe:MODulation:LFM:FSWeep	Sets or returns the Linear Frequency Modulation (LFM) Frequency Sweep method.
RADar:PULSe:MODulation:LFM:SRANge	Sets or returns the Linear Frequency Modulation (LFM) Sweep Range.
RADar:PULSe:MODulation:NLFM:BANDwidth	Sets or returns the Bandwidth for the Non Linear FM modulation.
RADar:PULSe:MODulation:NLFM:PROFile	Sets or returns the Profile for the Non Linear Frequency modulation.
RADar:PULSe:MODulation:NLFM:UDCPath	Sets or returns the filepath to the User Defined Coefficients profile for the Non Linear FM modulation.
RADar:PULSe:MODulation:P1Polyphase:CLENgth	Sets or returns the P1 Polyphase code length when the Modulation Phase is set to P1 Polyphase code.
RADar:PULSe:MODulation:P2Polyphase:CLENgth	Sets or returns the P2 Polyphase code length when the Modulation Phase is set to P2 Polyphase code.
RADar:PULSe:MODulation:P3Polyphase:CLENgth	Sets or returns the P3 Polyphase code length when the Modulation Phase is set to P3 Polyphase code.

Table 2-11: Pulse modulation commands and their descriptions (cont.)

Command	Description
RADar:PULSe:MODulation:P4Polyphase:CLENgth	Sets or returns the P4 Polyphase code length when the Modulation Phase is set to P4 Polyphase code.
RADar:PULSe:MODulation:PLFM:ADD	Adds the specified number of steps to the Piecewise Linear Frequency Modulation table.
RADar:PULSe:MODulation:PLFM:IOFFset	Sets or returns the Initial Offset (frequency) for the Piecewise Linear Frequency Modulation.
RADar:PULSe:MODulation:PLFM:STEP[n]:DELete	Deletes the specified step from the Piecewise Linear Frequency Modulation table.
RADar:PULSe:MODulation:PLFM:STEP[n]:DURation	Sets or returns the Duration of the specified step within the Piecewise Linear Frequency Modulation table.
RADar:PULSe:MODulation:PLFM:STEP[n]:FSWeep	Sets or returns the Frequency Sweep type of the specified step within the Piecewise Linear Frequency Modulation table.
RADar:PULSe:MODulation:PLFM:STEP[n]:SRANge	Sets or returns the Sweep Range of the specified step within the Piecewise Linear Frequency Modulation table.
RADar:PULSe:MODulation:POLYphase:IOFFset	Sets or returns the Initial Offset (initial phase offset) for the Polyphase Codes modulation.
RADar:PULSe:MODulation:POLYphase:POFFset	Sets or returns the Phase Offset for the Polyphase Codes modulation.
RADar:PULSe:MODulation:POLYphase:RAMP	Sets or returns the Ramp Function for the Polyphase Codes modulation.
RADar:PULSe:MODulation:POLYphase:RDURation	Sets or returns the Ramp Duration for the Polyphase Codes modulation.
RADar:PULSe:MODulation:POLYphase:STEPs	Sets or returns the Step Count for the Polyphase Codes modulation.
RADar:PULSe:MODulation:SFM:FOFFset	Sets or returns the Frequency Offset for the Step Frequency Modulation (SFM).
RADar:PULSe:MODulation:SFM:IOFFset	Sets or returns the Initial Frequency Offset for the Step Frequency Modulation (SFM).
RADar:PULSe:MODulation:SFM:RAMP	Sets or returns the Ramp Function type for the Step Frequency Modulation (SFM).
RADar:PULSe:MODulation:SFM:RDURation	Sets or returns the Ramp Duration for the Step Frequency Modulation (SFM).
RADar:PULSe:MODulation:SFM:STEPs	Sets or returns the number of steps (Step Count) for the Step Frequency Modulation (SFM).
RADar:PULSe:MODulation:UDCHirp:COUNt	Sets or returns the Up-Down Count for the Up-Down Chirp modulation.
RADar:PULSe:MODulation:UDCHirp:INVert	Sets or returns Invert state (enabled or disabled) of the Up-Down Chirp modulation.
RADar:PULSe:MODulation:UDCHirp:SRANge	Sets or returns the Sweep Range for the Up-Down Chirp modulation.
RADar:PULSe:MODulation:USFMam:ADD	Adds the specified number of steps to the User Defined Step FM AM Modulation table.

Table 2-11: Pulse modulation commands and their descriptions (cont.)

Command	Description
RADar:PULSe:MODulation:USFMam:RAMP	Sets or returns the Ramp Function type for the User Defined Step FM AM Modulation.
RADar:PULSe:MODulation:USFMam:RDURation	Sets or returns the Ramp Duration for the User Defined Step FM AM Modulation.
RADar:PULSe:MODulation:USFMam:STEP[n]:AMPLitude	Sets or returns the Amplitude of the specified step for the User Defined Step FM AM Modulation table.
RADar:PULSe:MODulation:USFMam:STEP[n]:DELete	Deletes the specified step from the User Defined Step FM AM Modulation table.
RADar:PULSe:MODulation:USFMam:STEP[n]:DURation	Sets or returns the Duration of the specified step within the User Defined Step FM AM Modulation table.
RADar:PULSe:MODulation:USFMam:STEP[n]:FOFFset	Sets or returns the Frequency Offset of the specified step for the User Defined Step FM AM Modulation table.
RADar:PULSe:MODulation:USPMam:ADD	Adds the specified number of steps to the User Defined Step PM AM Modulation table.
RADar:PULSe:MODulation:USPMam:RAMP	Sets or returns the Ramp Function type for the User Defined Step PM AM Modulation.
RADar:PULSe:MODulation:USPMam:RDURation	Sets or returns the Ramp Duration for the User Defined Step PM AM Modulation.
RADar:PULSe:MODulation:USPMam:STEP[n]:AMPLitude	Sets or returns the Amplitude of the specified step for the User Defined Step PM AM Modulation table.
RADar:PULSe:MODulation:USPMam:STEP[n]:DELete	Deletes the specified step from the User Defined Step PM AM Modulation table.
RADar:PULSe:MODulation:USPMam:STEP[n]:DURation	Sets or returns the Duration of the specified step within the User Defined Step PM AM Modulation table.
RADar:PULSe:MODulation:USPMam:STEP[n]:POFFset	Sets or returns the Phase Offset of the specified step for the User Defined Step PM AM Modulation table.

Pulse repetition commands

Table 2-12: Pulse repetition commands and their descriptions

Command	Description
RADar:PULSe:SPRI:DTYPe	Sets or returns the Staggered Pulse Repetition Interval deviation type.
RADar:PULSe:SPRI:FILE	Sets or returns the filepath to the Staggered PRI Deviation file.
RADar:PULSe:SPRI:RAMP:DEGRees	Sets or returns the Ramp Degrees for the PRI Deviation slope.
RADar:PULSe:SPRI:RAMP:MDEViation	Sets or returns the ramp Min Deviation (ΔT) for the PRI Deviation slope.
RADar:PULSe:SPRI:RAMP:SLOPe	Sets or returns the ramp Slope direction for the Staggered PRI Deviation ramp.

Table 2-12: Pulse repetition commands and their descriptions (cont.)

Command	Description
RADar:PULSe:SPRI:REPeat	Sets or returns the Repeat deviation pattern (enabled or disabled) for the Staggered PRI.
RADar:PULSe:SPRI:TABLe:ADD	Adds the specified number of repeat indices to the end of the user defined table of Staggered PRI.
RADar:PULSe:SPRI:TABLe:COUNt?	Returns the number of repeat indices in the user defined table of Staggered PRI.
RADar:PULSe:SPRI:TABLe:INDex[n]:DELete	Deletes the specified number of repeat indices, starting with the specified index location in the user defined table of Staggered PRI.
RADar:PULSe:SPRI:TABLe:INDex[n]:DEViation	Sets or returns the Deviation of the specified index location in the user defined table of Staggered PRI.
RADar:PULSe:SPRI:TABLe:INDex[n]:INSert	Inserts the specified number of repeat indices above the specified index location in the user defined table of Staggered PRI.
RADar:PULSe:SPRI:TON	Sets or returns the Staggered Pulse Repetition Interval state (enabled or disabled).

Pulse train commands

Table 2-13: Pulse train commands and their descriptions

Command	Description
RADar:PTRain:ADD	Creates and adds the named pulse train to the Pulse Train List.
RADar:PTRain:AMPLitude	Sets or returns the peak to peak baseband amplitude (in volts) for the selected pulse train.
RADar:PTRain:CARRier:COHerent	Sets or returns the Coherent Carrier state (enabled or disabled) for the selected pulse train.
RADar:PTRain:CARRier:FREQuency	Sets or returns the RF/IF carrier Frequency for the selected pulse train.
RADar:PTRain:CARRier:MAGNitude	Sets or returns the RF/IF carrier peak Magnitude (in dBm) for the selected pulse train.
RADar:PTRain:COUNt?	Returns the number of pulse train definitions in the Pulse Train List.
RADar:PTRain:DELete	Deletes the specified pulse train from the Pulse Train List.
RADar:PTRain:LSParameters	Sets or returns the pulse train S-Parameters link state (enabled or disabled) for the selected pulse train.
RADar:PTRain:OPEN	Imports a pulse train file.
RADar:PTRain:POFFset	Sets or returns the Phase Offset value for the selected pulse train.
RADar:PTRain:REName	Renames an existing pulse train name to a new name.
RADar:PTRain:SELect	Sets or returns the selected pulse train in the Pulse Train List.

Table 2-13: Pulse train commands and their descriptions (cont.)

Command	Description
RADar:PULSe:ADD	Adds a new pulse index to the selected pulse train in the Pulse Train List.
RADar:PULSe:COUNt?	This command returns the number of pulse definitions of the currently selected pulse train.
RADar:PULSe:CPGRoup:ADD	Adds a new pulse index to the selected combined pulse group of the selected pulse train.
RADar:PULSe:CPGRoup:COUNt?	Returns the number of pulse definitions in the selected combined pulse group.
RADar:PULSe:CPGRoup:DELete	Deletes the specified pulse index from the selected combined pulse group of the selected pulse train.
RADar:PULSe:CPGRoup:REPeat	Sets or returns the Repeat Count of the currently selected combined pulse group definition.
RADar:PULSe:CPGRoup:SELect	Sets or returns the index of the currently selected combined pulse group definition.
RADar:PULSe:CPGRoup:STIMe	Sets or returns the Start Time of the currently selected combined pulse group definition.
RADar:PULSe:DELete	Deletes the specified pulse index from the selected pulse train in the Pulse Train List.
RADar:PULSe:DTIMe:DURation	Sets or returns the Dead Time duration (off time) for the selected dead time pulse definition.
RADar:PULSe:REPeat	Sets or returns the Repeat Count of the currently selected pulse definition.
RADar:PULSe:SELect	Sets or returns the index of the currently selected pulse definition in the currently selected pulse train.
RADar:PULSe:STIMe	Sets or returns the Start Time of the currently selected pulse definition.
RADar:PULSe:TYPe	Sets or returns the type of the pulse definition for the selected pulse definition.

S-Parameters commands

Table 2-14: S-Parameters commands and their descriptions

Command	Description
RADar:SPARameter:MODE	Sets or returns the S-Parameter mode (Cascading or Non-Cascading).
RADar:SPARameter:SFORmat	Sets or returns the currently used signal format for all S-Parameter values.
RADar:SPARameter:SFORmat:LIQ	Sets or returns the Couple Settings (I,Q) state. When enabled, all I and Q S-Parameters are linked together (chained) so that all parameters match between I and Q.

Table 2-14: S-Parameters commands and their descriptions (cont.)

Command	Description
RADar:SPARameter:TON	Sets or returns the S-Parameter state (enabled or disabled).
RADar:SPARameter:BANDwidth	Sets or returns the S-Parameter bandwidth when setting manually.
RADar:SPARameter:BANDwidth:AUTO	Sets or returns the S-Parameter automatic bandwidth calculation setting.
RADar:SPARameter:CASCading:AGGRessor2[:ENABle]	Sets or returns the aggressor 2 signal type state (enabled or disabled) in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:AMPLitude	Sets or returns the specified Aggressor's amplitude, in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:CTALk	Sets or returns the specified Aggressor's crosstalk type, in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:DRATe	Sets or returns the specified Aggressor's data rate, in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal	Sets or returns specified Aggressor's signal type, in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE	Sets or returns the filepath to the aggressor file for the specified Aggressor, in Cascading mode.
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:PRBS	Sets or returns the specified Aggressor's PRBS signal type, in Cascading mode.
RADar:SPARameter:CASCading:DEEMbed	Sets or returns whether the Cascading S-Parameters is to de-embed (invert) the S-Parameters, in Cascading mode.
RADar:SPARameter:CASCading:STAGe[m]:DRX[n]	Sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified receiver port number (Rx-Port) in Cascading mode and Differential Signalling Scheme (where applicable).
RADar:SPARameter:CASCading:STAGe[m]:DTX[n]	Sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified transmission port number (Tx-Port) in Cascading mode and Differential Signalling Scheme (where applicable).
RADar:SPARameter:CASCading:STAGe[m][:ENABle]	Sets or returns the state of the specified Cascaded S-Parameter stage (enabled or disabled).
RADar:SPARameter:CASCading:STAGe[m]:FILE	Sets or returns the filepath for the specified S-Parameters Cascading Stage, in Cascading mode.
RADar:SPARameter:CASCading:STAGe[m]:RX[n]	Sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified receiver port number (Rx-Port) in Cascading mode and Single-Ended Signalling Scheme (where applicable).
RADar:SPARameter:CASCading:STAGe[m]:SSCHeme	Sets or returns the S-Parameter Signalling Scheme, in Cascading mode.
RADar:SPARameter:CASCading:STAGe[m]:TX[n]	Sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified transmission port number (Tx-Port) in Cascading mode and Single-Ended Signalling Scheme (where applicable).

Table 2-14: S-Parameters commands and their descriptions (cont.)

Command	Description
RADar:SPARameter:CASCading:STYPe	Sets or returns S-Parameter signal type (victim or aggressor), in Cascading mode.
RADar:SPARameter:CASCading:TYPE	Sets or returns the S-Parameter number of ports, in Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor2[:ENABle]	Sets or returns the aggressor 2 signal type state (enabled or disabled) in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:AMPLitude	Sets or returns the specified Aggressor's amplitude, in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:CTALk	Sets or returns the specified Aggressor's crosstalk type, in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:DRATe	Sets or returns the specified Aggressor's data rate, in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal	Sets or returns specified Aggressor's signal type, in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:FILE	Sets or returns the filepath to the aggressor file for the specified Aggressor, in Non-Cascading mode.
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:PRBS	Sets or returns the specified Aggressor's PRBS signal type, in Non-Cascading mode.
RADar:SPARameter:NCAScading:DEEMbed	Sets or returns whether the Non-Cascading S-Parameters is to de-embed (invert) the S-Parameters, in Non-Cascading mode.
RADar:SPARameter:NCAScading:DRX[n]	Sets or returns the S-Parameter port assignment of the channel's specified receiver port number (Rx-Port) in Non-Cascading mode and Differential Signalling Scheme (where applicable).
RADar:SPARameter:NCAScading:DTX[n]	Sets or returns the S-Parameter port assignment of the channel's specified transmission port number (Tx-Port) in Non-Cascading mode and Differential Signalling Scheme (where applicable).
RADar:SPARameter:NCAScading:FILE	Sets or returns the filepath and file name of the S-Parameter file, in Non-Cascading mode.
RADar:SPARameter:NCAScading:LAYout	Sets or returns the 4 port S-Parameter Matrix Configuration, in Non-Cascading mode.
RADar:SPARameter:NCAScading:RX[n]	Sets or returns the S-Parameter port assignment of the channel's specified receiver port number (Rx-Port) in Non-Cascading mode and Single-Ended Signalling Scheme (where applicable).
RADar:SPARameter:NCAScading:SSCHeme	Sets or returns the S-Parameter Signalling Scheme, in Non-Cascading mode.
RADar:SPARameter:NCAScading:STYPe	Sets or returns S-Parameter signal type (victim or aggressor), in Non-Cascading mode.
RADar:SPARameter:NCAScading:TX[n]	Sets or returns the S-Parameter port assignment of the channel's specified transmission port number (Tx-Port) in Non-Cascading mode and Single-Ended Signalling Scheme (where applicable).
RADar:SPARameter:NCAScading:TYPE	Sets or returns the S-Parameter number of ports, in Non-Cascading mode.

Commands in alphabetical order

This section contains all available commands. They are presented in alphabetical order.

Use the Command Groups section to simplify navigating to specific groups of commands.

RADar:COMPile (No Query Form)

This command compiles and generates waveforms and sequences using the Radar plug-in compile settings.

Conditions This is an overlapping command. Overlapping commands run concurrently with other commands, allowing additional commands to start before the overlapping command has finished.

Group Compile

Syntax `RADar:COMPile`

Examples `RADAR:COMPILE` compiles and generates the waveforms/sequences.

RADar:COMPile:CANCel (No Query Form)

This command cancels a compilation currently in progress.

Group Compile

Syntax `RADar:COMPile:CANCel`

Examples `RADAR:COMPILE:CANCEL` immediately ends the current compile process.

RADar:COMPile:CASSign

This command sets or returns the state (enabled or disabled) to compile the waveform and immediately assign it to a specified channel (enabled) or just compile the waveform (disabled).

Group Compile

Syntax

```
RADar:COMPile:CASSign {0|1|OFF|ON}
RADar:COMPile:CASSign?
```

Related Commands RADar:PTRain:CHANnel:I, RADar:PTRain:CHANnel:Q, RADar:PTRain:CHANnel:RF, RADar:COMPile:PLAY

Arguments 0 or OFF will only compile the waveform. OFF or 0 is the default value.
1 or ON will compile and assign the waveform to a channel.

Returns A single <Boolean> value.

Examples `RADAR:COMPILE:CASSIGN 1` enables the compile and assign function.

`RADAR:COMPILE:CASSIGN?` might return `0`, indicating that the compile and assign function is disabled.

RADar:COMPile:OSAMpling

This command sets or returns the over sampling rate used to determine the sampling rate of the compiled signal.

The Sampling Rate is calculated by multiplying the Over sampling value with the maximum frequency of the signal to be generated.

Group Compile

Syntax

```
RADar:COMPile:OSAMpling <Oversampling>
RADar:COMPile:OSAMpling?
```

Arguments Oversampling>::=<NR1> value.
Range: 1 to 1000

Returns A single <NR1> value.

Examples `RADAR:COMPILE:OSAMPLING 6` sets the over sampling rate to 6.

`RADAR:COMPILE:OSAMPLING?` might return `4`, indicating that the over sampling rate is set to 4.

RADar:COMPile:OVERwrite

This command sets or returns the Overwrite existing waveforms/sequences state (enabled or disabled) when compiling. When enabled, new waveforms and sequences overwrite existing waveforms and sequences of the same name. When disabled, new waveforms are generated with an additional numeric suffix added at the end of each name.

Group Compile

Syntax

```
RADar:COMPile:OVERwrite {1|0|ON|OFF}
RADar:COMPile:OVERwrite?
```

Arguments ON or 1 enables Overwriting names. ON or 1 is the default value.
OFF or 0 disables Overwriting names.

Returns A single <Boolean> value.

Examples `RADAR:COMPILE:OVERWRITE 1` enables Overwriting names.

`RADAR:COMPILE:OVERWRITE?` might return `0`, indicating that Overwriting names is disabled.

RADar:COMPile:PLAY

This command sets or returns the Play after assign state (enabled or disabled) for the Compile Settings.

Play after assign is active only when Compile and assign is enabled.

Group Compile

Syntax

```
RADar:COMPile:PLAY {0|1|OFF|ON}
RADar:COMPile:PLAY?
```

Arguments OFF or 0 disables Play after assign. OFF or 0 is the default value.

ON or 1 enables Play after assign.

Returns A single <Boolean> value.

Examples `RADAR:COMPILE:PLAY 1` enables Play after assign.

`RADAR:COMPILE:PLAY?` might return `0`, indicating that Play after assign is disabled.

RADar:COMPile:SEQuence

This command sets or returns the state (enabled or disabled) to create sequences at end of the compile. If the sequence status is off, then each pulse train is created as one single waveform.

Conditions The instrument must have sequencing available to enable creating sequences.

Group Compile

Syntax

```
RADar:COMPile:SEQuence {0|1|OFF|ON}
RADar:COMPile:SEQuence?
```

Arguments ON or 1 enables creating . ON or 1 is the default value.

OFF or 0 disables Play after assign.

Returns A single <Boolean> value.

Examples `RADAR:COMPILE:SEQUENCE 1` enables creating sequences.

`RADAR:COMPILE:SEQUENCE?` might return `0`, indicating that creating sequences is disabled.

RADar:COMPile:SRATe

This command sets or returns the manual Sampling Rate for the compile settings.

Group Compile

Syntax

```
RADar:COMPile:SRATe <rate>
RADar:COMPile:SRATe?
```

Arguments <rate>::= <NRf> value.

The maximum sampling rate is dependent on the instrument and instrument options.

Returns A single <NR3> value.

Examples `RADAR:COMPILE:SRATE 1E9` sets the Sampling Rate to 1 GS/s in the Compile settings.

`RADAR:COMPILE:SRATE?` might return `2.0000000000E+9`, indicating that the sampling rate is 2 GS/s

RADar:COMPile:SRATe:AUTO

This command sets or returns the sampling rate Auto Calculate state (enabled or disabled). When enabled, the sampling rate is automatically calculated at compile time.

Group Compile

Syntax

```
RADar:COMPile:SRATe:AUTO {0|1|OFF|ON}
RADar:COMPile:SRATe:AUTO?
```

Arguments OFF or 0 disables the sampling rate auto calculation and the manual setting is used.

ON or 1 enables the auto calculation. ON or 1 is the default value.

Returns A single <Boolean> value.

Examples `RADAR:COMPILE:SRATE:AUTO ON` sets the signal's sampling rate to be automatically calculated at compile time.

`RADAR:COMPILE:SRATE:AUTO?` might return `0`, indicating that the sampling rate will not be automatically calculated at compile time.

RADar:PTRain:ADD (No Query Form)

This command creates and adds the named pulse train to the Pulse Train List. The pulse train is created using the default settings and is added to the end of the existing list of pulse trains.

The Pulse Train List can contain a maximum of 10 pulse train definitions.

Group Pulse train

Syntax `RADar:PTRain:ADD <pulse_train_name>`

Arguments <pulse_train_name>:= <string>

Examples `RADAR:PTRAIN:ADD "Sim_1"` adds a new pulse train named "Sim_1" to the end of Pulse Train List.

RADar:PTRain:AMPLitude

This command sets or returns the peak to peak baseband amplitude (in volts) for the selected pulse train.

Conditions Signal format must be set to IQ.

Group Pulse train

Syntax

```
RADar:PTRain:AMPLitude <amplitude>
RADar:PTRain:AMPLitude?
```

Related Commands

Arguments <amplitude>::=<NRf> value.
Range: 25 mV_{pp} to 1.5 V_{pp}.
Default: 632 mV_{pp}.

Returns A single <NR3> value.

Examples `RADAR:PTRAIN:AMPLITUDE 250E-3` sets the baseband Amplitude to 250 mV_{pp}.

`RADAR:PTRAIN:AMPLITUDE?` might return `25.0000000000E-3`, indicating that the baseband Amplitude is set to 25 mV_{pp}.

RADar:PTRain:CARRier:COHerent

This command sets or returns the Coherent Carrier state (enabled or disabled) for the selected pulse train. When enabled, the carrier as a coherent carrier. This is not available for an IQ signal.

Conditions Signal format must be set to RF/IF.

Group Pulse train

Syntax

```
RADar:PTRain:CARRier:COHerent {1|0|ON|OFF}
RADar:PTRain:CARRier:COHerent?
```

Arguments ON or 1 enables Coherent Carrier. ON or 1 is the default value.

OFF or 0 disables Coherent Carrier.

Returns A single <Boolean> value.

Examples `RADAR:PTRAIN:CARRIER:COHERENT 1` enables Coherent Carrier for the selected pulse train.

`RADAR:PTRAIN:CARRIER:COHERENT?` might return `0`, indicating that Coherent Carrier is disabled for the selected pulse train.

RADar:PTRain:CARRier:FREQuency

This command sets or returns the RF/IF carrier Frequency for the selected pulse train.

Conditions Signal format must be set to RF/IF.

Group Pulse train

Syntax

```
RADar:PTRain:CARRier:FREQuency <Carrier_Frequency>
RADar:PTRain:CARRier:FREQuency?
```

Arguments <Carrier_Frequency>::=<NRf> value.
Range: 100 Hz to 2.5 GHz.
Default: 100 MHz.

Returns <Carrier_Frequency>::=<NR3> value.

Examples `RADAR:PTRAIN:CARRIER:FREQUENCY 1E9` sets the carrier Frequency to 1 GHz.

`RADAR:PTRAIN:CARRIER:FREQUENCY?` might return `1.0000000000E+9`, indicating that the carrier Frequency is set to 1 GHz.

RADar:PTRain:CARRier:MAGNitude

This command sets or returns the RF/IF carrier peak Magnitude (in dBm) for the selected pulse train.

Conditions Signal format must be set to RF/IF.

Group Pulse train

Syntax

```
RADar:PTRain:CARRier:MAGNitude <peak_Magnitude>
RADar:PTRain:CARRier:MAGNitude?
```

Arguments <peak_Magnitude>::=<NRf> value.
Range: –28.06 dBm to 7.5 dBm.
Default: 0 dBm.

Returns <peak_Magnitude>::=<NR3> value.

Examples `RADAR:PTRAIN:CARRIER:MAGNITUDE −2` sets the carrier peak Magnitude to –2 dBm.

`RADAR:PTRAIN:CARRIER:MAGNITUDE?` might return `5.0000000000`, indicating that the carrier peak Magnitude is set to 5 dBm.

RADar:PTRain:CHANnel:I

This command sets or returns the playout channel intended for the compiled I signal waveform when the signal format is set to IQ. The selected channel is also used to define the amplitude ranges.

If RADar:COMPile:CASSign is enabled, the waveform is assigned to the specified channel.

Group Compile

Syntax

```
RADar:PTRain:CHANnel:I <channel>
RADar:PTRain:CHANnel:I?
```

Related Commands RADar:PTRain:CHANnel:Q, RADar:COMPile:CASSign, RADar:COMPile:PLAY

Arguments <channel>::=<NR1> is a valid channel number.

Returns A single <NR1> value.

Examples `RADAR:PTRAIN:CHANNEL:I 2` assigns the I signals to channel 2.

`RADAR:PTRAIN:CHANNEL:I?` might return `1`, indicating that I signals are assigned to channel 1.

RADar:PTRain:CHANnel:Q

This command sets or returns the playout channel intended for the compiled Q signal waveform when the signal format is set to IQ. The selected channel is also used to define the amplitude ranges.

If RADar:COMPile:CASSign is enabled, the waveform is assigned to the specified channel.

Group Compile

Syntax

```
RADar:PTRain:CHANnel:Q <channel>
RADar:PTRain:CHANnel:Q?
```

Related Commands RADar:PTRain:CHANnel:I, RADar:COMPile:CASSign, RADar:COMPile:PLAY

Arguments <channel>::=<NR1> is a valid channel number.

Returns A single <NR1> value.

Examples `RADAR:PTRAIN:CHANNEL:Q 2` assigns the Q signals to channel 2.

`RADAR:PTRAIN:CHANNEL:Q?` might return `1`, indicating that Q signals are assigned to channel 1.

RADar:PTRain:CHANnel:RF

This command sets or returns the playout channel intended for the compiled RF waveforms and sequences when the signal format is set to RF/IF. The selected channel is also used to define the amplitude ranges.

If RADar:COMPile:CASSign is enabled, the waveform is assigned to the specified channel.

Group Compile

Syntax

```
RADar:PTRain:CHANnel:RF <channel>
RADar:PTRain:CHANnel:RF?
```

Related Commands RADar:COMPile:CASSign, RADar:COMPile:PLAY

Arguments <channel>::=<NR1> is a valid channel number.

Channel 1 is the default channel.

Returns A single <NR1> value.

Examples `RADAR:PTRAIN:CHANNEL:RF 2` compiles RF waveforms for channel 2.

`RADAR:PTRAIN:CHANNEL:RF?` might return `1`, indicating that RF are waveforms compiled for channel 1.

RADar:PTRain:CORRection:APPLy

This command sets or returns the Apply Corrections File state (enabled or disabled) for the Compile Settings.

When applying correction files, you have the capability adjust the frequency response.

- When applying an RF correction file, you can apply a Gaussian filter/bandwidth or remove Sin(x)/x distortions.
- When applying an I/Q correction file, you can apply a skew.

These settings are global settings controlled by the host instrument. The PI commands to access these settings are provided in the host instrument's programmer manual.

Group Compile

Syntax

```
RADar:PTRain:CORRection:APPLy {1|0|ON|OFF}
RADar:PTRain:CORRection:APPLy?
```

Related Commands RADar:PTRain:CORRection:PATH

Arguments OFF or 0 disables Apply Corrections File. OFF or 0 is the default value.

ON or 1 enables Apply Corrections File.

Returns A single <Boolean> value.

Examples `RADAR:PTRAIN:CORRECTION:APPLY ON` enables Apply Corrections File.

`RADAR:PTRAIN:CORRECTION:APPLY?` might return `0`, indicating that Apply Corrections File is disabled.

RADar:PTRain:CORRection:PATH

This command sets or returns the RF Correction filename and filepath to use when compiling an RF signal.

Conditions The signal format must be set to RF.

The Precompensation plug-in is required.

Group Compile

Syntax

```
RADar:PTRain:CORRection:PATH <filepath>
RADar:PTRain:CORRection:PATH?
```

Related Commands RADar:PTRain:CORRection:APPLy

Arguments <filepath>::=<string> defines path to the correction file.

Returns A single <filepath> string.

Examples `RADAR:PTRAIN:CORRECTION:PATH "C:\temp\CorrectionFile.corr"` sets the RF Correction filename and filepath.

`RADAR:PTRAIN:CORRECTION:PATH?` might return `"C:\temp\CorrectionFile.corr"` as the RF Correction filename and filepath.

RADar:PTRain:CORRection:PATH:I

This command sets or returns the I Correction filename and filepath to use when compiling and IQ signal.

Conditions The signal format must be set to IQ.

Apply corrections must be enabled.

Group Compile

Syntax `RADar:PTRain:CORRection:PATH:I <filepath>`

Related Commands RADar:PTRain:CORRection:TYPE, RADar:PTRain:CORRection:APPLy

Arguments <filepath>::=<string> defines path to the I correction file.

Returns A single <filepath> string.

Examples `RADAR:PTRAIN:CORRECTION:PATH:I "C:\temp\ICorrectionFile.corr"` sets the I Correction filename and filepath.

`RADAR:PTRAIN:CORRECTION:PATH:I?` might return `"C:\temp\ICorrectionFile.corr"` as the I Correction filename and filepath.

OFDM:PTRain:CORRection:PATH:IQ

This command sets or returns the IQ Correction filename and filepath to use when compiling and IQ signal.

Conditions The signal format must be set to IQ.

Apply corrections must be enabled.

Group Compile

Syntax `OFDM:PTRain:CORRection:PATH:IQ <filepath>`

Related Commands RADar:PTRain:CORRection:TYPE, RADar:PTRain:CORRection:APPLy

Arguments <filepath>::=<string> defines path to the IQ correction file.

Returns A single <filepath> string.

Examples `OFDM:PTRAIN:CORRECTION:PATH:IQ "C:\temp\ICorrectionFile.corr"` sets the IQ Correction filename and filepath.

`OFDM:PTRAIN:CORRECTION:PATH:IQ?` might return `"C:\temp\ICorrectionFile.corr"` as the IQ Correction filename and filepath.

RADar:PTRain:CORRection:PATH:Q

This command sets or returns the Q correction filename and filepath to use when compiling and IQ signal.

Conditions The signal format must be set to IQ.

Apply corrections must be enabled.

Group Compile

Syntax `RADar:PTRain:CORRection:PATH:Q <filepath>`

Related Commands RADar:PTRain:CORRection:TYPE, RADar:PTRain:CORRection:APPLy

Arguments <filepath>::=<string> defines path to the Q correction file.

Returns A single <filepath> string.

Examples `RADAR:PTRAIN:CORRECTION:PATH:Q "C:\temp\QCorrectionFile.corr"` sets the Q Correction filename and filepath.

`RADAR:PTRAIN:CORRECTION:PATH:Q?` might return `"C:\temp\QCorrectionFile.corr"` as the Q Correction filename and filepath.

RADar:PTRain:CORRection:TYPE

This command sets or returns the type of IQ correction file (a single IQ file or individual I and Q files) to apply when compiling.

Conditions The signal format must be set to IQ.

Group Compile

Syntax

```
RADar:PTRain:CORRection:TYPE {IQ|BOTH}
RADar:PTRain:CORRection:TYPE?
```

Arguments IQ: The compile process uses a single IQ correction file.

BOTH: The compile process uses both an I correction file and a Q correction file.

Returns IQ|BOTH

Examples `RADAR:PTRAIN:CORRECTION:TYPE IQ` sets the compiler to apply a single IQ correction file when compiling.

`RADAR:PTRAIN:CORRECTION:TYPE?` might return `BOTH`, indicating that the compiler will apply both an I and Q correction file to the I and Q signals.

RADar:PTRain:DUPConverter

This command sets or returns the Internal IQ Modulator state (enabled or disabled) when compiling.

When enabled, a complex IQ Waveform is created which can be used with the internal IQ modulator. Sampling Rate and interpolation rates will be calculated based on the Baseband parameters.

When disabled, a separate I and Q signals are created.

Conditions The instrument must have an internal IQ Modulator and the Signal Format set to IQ.

Group Compile

Syntax

```
RADar:PTRain:DUPConverter {0|1|OFF|ON}
RADar:PTRain:DUPConverter?
```

Arguments OFF or 0 disables the internal IQ Modulator. OFF or 0 is the default value.

ON or 1 enables the internal IQ Modulator.

Returns A single <Boolean> value.

Examples `RADAR:PTRAIN:DUPCONVERTER ON` enables the internal IQ Modulator.

`RADAR:PTRAIN:DUPCONVERTER?` might return `0`, indicating the internal IQ Modulator is disabled.

RADar:PTRain:COUNt? (Query Only)

This command returns the number of pulse train definitions in the Pulse Train List.

Group Pulse train

Syntax `RADar:PTRain:COUNt?`

Returns A single <NR1> value.

Examples `RADAR:PTRAIN:COUNT?` might return `3.0000000000`, indicating that there are 3 pulse train definitions in the Pulse Train List.

RADar:PTRain:DELete (No Query Form)

This command deletes the specified pulse train from the Pulse Train List.

The Pulse Train List must contain at least one pulse train definition.

Group Pulse train

Syntax `RADar:PTRain:DELete <pulse_train_name>`

Arguments <pulse_train_name>:= <string>

Examples `RADAR:PTRAIN:DELETE "Sim_1"` deletes the pulse train named "Sim_1" from the Pulse Train List.

RADar:PTRain:LSParameters

This commend sets or returns the pulse train S-Parameters link state (enabled or disabled) for the selected pulse train.

When enabled, all pulse train S-Parameters are linked together so that all parameters match.

Group Pulse train

Syntax

```
RADar:PTRain:LSParameters {1|0|ON|OFF}
RADar:PTRain:LSParameters?
```

Arguments OFF or 0 disables linking all pulse train S-Parameters together. OFF or 0 is the default value.

ON or 1 enables linking all pulse train S-Parameters together.

Returns A single <Boolean> value.

Examples `RADAR:PTRAIN:LSPARAMETERS 1` enables linking all pulse train S-Parameters together for the selected pulse train.

`RADAR:PTRAIN:LSPARAMETERS?` might return `0`, indicating that linking the pulse train S-Parameters is disabled for the selected pulse train.

RADar:PTRain:OPEN (No Query Form)

This command imports a pulse train file. The complete path to the file and filename must be provided.

Only comma separated files (.csv) are acceptable.

Group Pulse train

Syntax `RADar:PTRain:OPEN <csv_filepath>`

Arguments <csv_filepath> := <string> defines the path and pulse train file.

Examples `RADAR:PTRAIN:OPEN "C:\sim1.csv"` imports the pulse train file sim1.csv from the path provided.

RADar:PTRain:POFFset

This command sets or returns the Phase Offset value for the selected pulse train.

Group Pulse train

Syntax

```
RADar:PTRain:POFFset <phase_offset>
RADar:PTRain:POFFset?
```

Arguments <phase_offset>::=<NRf> value.
Range: –180 ° to +180 °.

Default: 0.

Returns A single <NR3> value.

Examples `RADAR:PTRAIN:POFFSET 10` sets the Phase Offset to 10 ° for the selected pulse train.

`RADAR:PTRAIN:POFFSET?` might return `22.0000000000`, indicating that the Phase Offset is set to 22 ° for the selected pulse train.

RADar:PTRain:REName (No Query Form)

This command renames an existing pulse train name to a new name. The pulse train being renamed must exist in the Pulse Train List.

Group Pulse train

Syntax `RADar:PTRain:REName <Old_name>,<New_name>`

Arguments <Old_name>::=<string> defines the pulse train to be renamed.
<New_name>::=<string> defines the new pulse train name.

Examples `RADAR:PTRAIN:RENAME "PulseTrain1",PulseTrain1new"` renames the frame PulseTrain1 to PulseTrain1new.

RADar:PTRain:SELect

This command sets or returns the selected pulse train in the Pulse Train List.

Other commands act on the selected pulse train.

Group Pulse train

Syntax

```
RADar:PTRain:SELect <pulse_train_name>
RADar:PTRain:SELect?
```

Arguments <pulse_train_name>:= <string>

Returns A single <pulse_train_name> string.

Examples `RADAR:PTRAIN:SELECT "Sim_1"` selects the pulse train named `Sim_1` in the Pulse Train List.

`RADAR:PTRAIN:SELECT?` might return `Sim_1` as the selected pulse train in the Pulse Train List.

RADar:PULSe:ADD (No Query Form)

This command adds a new pulse index to the selected pulse train in the Pulse Train List.

Group Pulse train

Syntax `RADar:PULSe:ADD {CPGRoup|PULSed|DTIMe}`

Arguments CPGRoup: Selects the type Combined Pulse Group, allowing multiple sub-pulse groups.
PULSed: Selects the type Pulse Group.
DTIMe: Selects the Dead Time type.

Returns CPGR: Combined Pulse Group.
PULS: Pulse Group.
DTIM: Dead Time.

Examples `RADAR:PULSE:ADD DTIMe` adds a dead time pulse index to the currently selected pulse train.

RADar:PULSe:ANTenna:BEAM:AZIMuth

This command sets or returns the antenna Beam Azimuth for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:BEAM:AZIMuth <beam_azimuth>
RADar:PULSe:ANTenna:BEAM:AZIMuth?
```

Arguments <beam_azimuth>::= <NRf> value.
Range: 0.01° to 360°.
Default: 4°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:ANTENNA:BEAM:AZIMUTH 10.5` sets the antenna Beam Azimuth to 10.5 degrees for the selected pulse definition.

`RADAR:PULSE:ANTENNA:BEAM:AZIMUTH?` might return `1.0000000000`, indicating that the antenna Beam Azimuth is set to 1 degree for the selected pulse definition.

RADar:PULSe:ANTenna:BEAM:ELEVation

This command sets or returns the antenna Beam Elevation for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:BEAM:ELEVation <beam_elevation>
RADar:PULSe:ANTenna:BEAM:ELEVation?
```

Arguments <beam_elevation>::= <NRf> value.
Range: 0.01° to 360°.
Default: 1°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:ANTENNA:BEAM:ELEVATION 10.5` sets the antenna Beam Elevation to 10.5 degrees for the selected pulse definition.

`RADAR:PULSE:ANTENNA:BEAM:ELEVATION?` might return `1.0000000000`, indicating that the antenna Beam Elevation is set to 1 degree for the selected pulse definition.

RADar:PULSe:ANTenna:BEAM:TYPE

This command sets or returns the antenna Beam Type for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:BEAM:TYPE {SINC|GAUSsian}
RADar:PULSe:ANTenna:BEAM:TYPE?
```

Arguments

SINC: Sin(x) / x.
GAUSsian: Gaussian.

Returns

SINC
GAUS

Examples

`RADAR:PULSE:ANTENNA:BEAM:TYPE SINC` sets the antenna Beam Type to Sin(x) / x for the selected pulse definition.

`RADAR:PULSE:ANTENNA:BEAM:TYPE?` might return `GAUS`, indicating that the antenna Beam Type is set to Gaussian for the selected pulse definition.

RADar:PULSe:ANTenna:SCAN:AREA

This command sets or returns the antenna Scan Area (in degrees) for the selected pulse definition. Adjusting the Scan Area affects the Scan Rate setting.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:SCAN:AREA <scan_area>
RADar:PULSe:ANTenna:SCAN:AREA?
```

Related Commands RADar:PULSe:ANTenna:SCAN:RATE

Arguments <scan_area>::= <NRf> value.
Range: 0.0001012° to 100000000°.
Default: 180°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:ANTENNA:SCAN:AREA 100E3` sets the antenna scan area is set to 100000 degrees for the selected pulse definition.

`RADAR:PULSE:ANTENNA:SCAN:AREA?` might return `1.0000010000E+6`, indicating that the antenna scan area is set to 1000001 degrees for the selected pulse definition.

RADar:PULSe:ANTenna:SCAN:FILE

This command sets or returns the filepath to the custom antenna scan file for the selected pulse definition.

Refer to the Radar User Help for information to create a custom scan file.

Conditions Scan Type must be set to Custom.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:SCAN:FILE <filepath>
RADar:PULSe:ANTenna:SCAN:FILE?
```

Related Commands RADar:PULSe:ANTenna:SCAN:TYPE

Arguments <filepath>::= <string>.

Returns A single <filepath> string.

Examples `RADAR:PULSE:ANTENNA:SCAN:FILE "C:\ScanFile.txt"` sets the custom Scan filepath and filename for the selected pulse definition.

`RADAR:PULSE:ANTENNA:SCAN:FILE?` might return `"C:\ScanFile.txt"`, indicating the custom antenna Scan filepath and filename for the selected pulse definition.

RADar:PULSe:ANTenna:SCAN:RATE

This command sets or returns the antenna Scan Rate (in degrees/second) for the selected pulse definition. Adjusting the Scan Rate affects the Scan Area setting.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:SCAN:RATE <scan_rate>
RADar:PULSe:ANTenna:SCAN:RATE?
```

Related Commands RADar:PULSe:ANTenna:SCAN:AREA

Arguments <scan_rate>::= <NRf> value.
Range: 1 deg/s to 10 Mdeg/s.
Default: 180 deg/s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:ANTENNA:SCAN:RATE 10E3` sets the antenna Scan Rate to 10 kdeg/s for the selected pulse definition.

`RADAR:PULSE:ANTENNA:SCAN:RATE?` might return `1.0000000000E+6`, indicating that the antenna Scan Rate is set to 1 Mdeg/s for the selected pulse definition.

RADar:PULSe:ANTenna:SCAN:TYPE

This command sets or returns the antenna scan Type (antenna beam pattern) for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:SCAN:TYPE {CIRCular|CUSTom}
RADar:PULSe:ANTenna:SCAN:TYPE?
```

Arguments

CIRCular
CUSTom

Returns

CIRC
CUST

Examples

`RADAR:PULSE:ANTENNA:SCAN:TYPE CIRCULAR` sets the antenna scan type to Circular for the selected pulse definition.

`RADAR:PULSE:ANTENNA:SCAN:TYPE?` might return `CUST`, indicating that the scan type is set to Custom for the selected pulse definition.

RADar:PULSe:ANTenna:TARGet:HORizontal

This command sets or returns the antenna Horizontal Target position for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:TARGet:HORizontal <target_horizontal>
RADar:PULSe:ANTenna:TARGet:HORizontal?
```

Related Commands RADar:PULSe:ANTenna:TARGet:VERTical

Arguments <target_horizontal>::= <NRf> value.
Range: –180° to 180°.
Default: 180°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:ANTENNA:TARGET:HORIZONTAL –25` sets the antenna Horizontal Target position to –25 degrees for the selected pulse definition.

`RADAR:PULSE:ANTENNA:TARGET:HORIZONTAL?` might return `40.0000000000`, indicating that the antenna Horizontal Target position is set to 40 degrees for the selected pulse definition.

RADar:PULSe:ANTenna:TARGet:VERTical

This command sets or returns the antenna Vertical Target position for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:TARGet:VERTical <target_vertical>
RADar:PULSe:ANTenna:TARGet:VERTical?
```

Related Commands RADar:PULSe:ANTenna:TARGet:HORizontal

Arguments <target_vertical>::= <NRf> value.
Range: –180° to 180°.
Default: 180°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:ANTENNA:TARGET:VERTICAL –25` sets the antenna Vertical Target position to –25 degrees for the selected pulse definition.

`RADAR:PULSE:ANTENNA:TARGET:VERTICAL?` might return `40.0000000000`, indicating that the antenna Vertical Target position is set to 40 degrees for the selected pulse definition.

RADar:PULSe:ANTenna:TON

This command sets or returns the Antenna settings state (enabled or disabled) for the selected pulse definition.

Group Antenna

Syntax

```
RADar:PULSe:ANTenna:TON {1|0|ON|OFF}
RADar:PULSe:ANTenna:TON?
```

Arguments OFF or 0 disables the Antenna settings. OFF or 0 is the default value.

ON or 1 enables the Antenna settings.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:ANTENNA:TON 1` enables the Antenna settings for the selected pulse definition.

`RADAR:PULSE:ANTENNA:TON?` might return `0`, indicating that the Antenna settings are disabled for the selected pulse definition.

RADar:PULSe:COUNt? (Query Only)

This command returns the number of pulse definitions of the currently selected pulse train.

Group Pulse train

Syntax `RADar:PULSe:COUNt?`

Returns A single <NR1> value.

Examples `RADAR:PULSE:COUNT?` might return `6`, indicating that there are six pulse definitions in the currently selected pulse train.

RADar:PULSe:CPGRoup:ADD (No Query Form)

This command adds a new pulse index to the selected combined pulse group of the selected pulse train.

Conditions The selected pulse definition must be of the type Combined Pulse Group.

Group Pulse train

Syntax `RADar:PULSe:CPGRoup:ADD`

Examples `RADAR:PULSE:CPGROUP:ADD` creates a new pulse definition index to the selected Combined Pulse Group.

RADar:PULSe:CPGRoup:COUNt? (Query Only)

This command returns the number of pulse definitions in the selected combined pulse group.

Group Pulse train

Syntax `RADar:PULSe:CPGRoup:COUNt?`

Returns A single <NR1> value.

Examples `RADAR:PULSE:CPGROUP:COUNT?` might return `3`, indicating that there are 3 pulse definitions in the combined pulse group.

RADar:PULSe:CPGRoup:DELete (No Query Form)

This command deletes the specified pulse index from the selected combined pulse group of the selected pulse train.

Conditions The selected pulse definition must be of the type Combined Pulse Group.

Group Pulse train

Syntax `RADar:PULSe:CPGRoup:DELete <index>`

Arguments <index>::= <NR1> value (valid combined pulse index).

Examples `RADAR:PULSE:CPGROUP:DELETE 3` deletes the pulse at index 3 of the selected Combined Pulse Group.

RADar:PULSe:CPGRoup:REPeat

This command sets or returns the Repeat Count of the currently selected combined pulse group definition.

Group Pulse train

Syntax

```
RADar:PULSe:CPGRoup:REPeat <repeat_count>
RADar:PULSe:CPGRoup:REPeat?
```

Arguments <repeat_count>::= <NR1> value.
Default: 1.

Returns A single <NR1> value.

Examples `RADAR:PULSE:CPGROUP:REPEAT 10` sets the Repeat Count of the combined pulse group definition to 10.

`RADAR:PULSE:CPGROUP:REPEAT?` might return `5`, indicating that the Repeat Count of the combined pulse group definition is set to 5.

RADar:PULSe:CPGRoup:SELect

This command sets or returns the index of the currently selected combined pulse group definition. Other commands act on the selected combined pulse group.

Group Pulse train

Syntax

```
RADar:PULSe:CPGRoup:SELect <index>
RADar:PULSe:CPGRoup:SELect?
```

Arguments <index>::=<NR1> value is a valid pulse definition index in the combined pulse group.

Returns A single <NR1> value.

Examples `RADAR:PULSE:CPGROUP:SELECT 2` selects pulse definition index 2 of the currently selected combined pulse group definition.

`RADAR:PULSE:CPGROUP:SELECT?` might return 4, indicating that index four is the currently selected pulse definition of the selected combined pulse group definition.

RADar:PULSe:CPGRoup:STIMe

This command sets or returns the Start Time of the currently selected combined pulse group definition.

Group Pulse train

Syntax

```
RADar:PULSe:CPGRoup:STIMe <start_time>
RADar:PULSe:CPGRoup:STIMe?
```

Arguments <start_time>::= <NRf> value.
Range: 0 s to 1 s.
Default: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:CPGROUP:STIME 12E-6` sets the Start Time of the pulse definition to 12 µs.

`RADAR:PULSE:CPGROUP:STIME?` might return `1.0000000000E-12`, indicating that the Start Time of the pulse definition is set to 1 ps.

RADar:PULSe:DELete (No Query Form)

This command deletes the specified pulse index from the selected pulse train in the Pulse Train List.

Group Pulse train

Syntax `RADar:PULSe:DELete <index>`

Arguments <index>::= <NR1> value (valid pulse index).

Examples `RADAR:PULSE:DELETE 3` deletes the pulse at index 3 of the pulse list of the currently selected pulse train.

RADar:PULSe:DTIMe:DURation

This command sets or returns the Dead Time duration (off time) for the selected dead time pulse definition.

Conditions The pulse definition type must be set to Dead Time.

Group Pulse train

Syntax

```
RADar:PULSe:DTIMe:DURation <duration>
RADar:PULSe:DTIMe:DURation
```

Arguments <duration>::= <NRf> value.
Range: 0 s to 1 s.
Default: 100 µs.

Returns A single <NR3> value.

Examples `RADAR:PULSE:DTIME:DURATION 1E-2` sets the pulse envelope Dead Time Duration to 10 ms for the selected pulse definition.

`RADAR:PULSE:DTIME:DURATION?` might return `100.0000000000E-6`, indicating that the pulse envelope Dead Time Duration is set to 100 µs for the selected pulse definition.

RADar:PULSe:HOPPing:CUSTom:RLISt

This command sets or returns the Repeat List state (enabled or disabled) for the Frequency Hop table for the selected pulse definition. When enabled, the Frequency Hop List repeats when the Hop List has been completed.

The Hopping Pattern must be set to Custom and the Hop Time must be set to "Pulses Per Hop".

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:CUSTom:RLISt {1|0|ON|OFF}
RADar:PULSe:HOPPing:CUSTom:RLISt?
```

Arguments OFF or 0 disables Repeat List. OFF or 0 is the default value.

ON or 1 enables Repeat List.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:HOPPING:CUSTOM:RLIST 1` enables the Repeat List so that the Frequency Hop List repeats when the Hop List has been completed for the selected pulse definition.

`RADAR:PULSE:HOPPING:CUSTOM:RLIST?` might return `0`, indicating that the Repeat List is disabled so that the Frequency Hop List is not repeated for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:ADD (No Query Form)

This command adds a single entry to the end of the Frequency Hop List or the Frequency Avoid List for the selected pulse definition.

When the Hopping Pattern is either Custom or Pseudo Random List, the Frequency Hop List modified. When the Hopping Pattern is set to Pseudo Random Range, the Frequency Avoid List is modified.

Group Hopping

Syntax `RADar:PULSe:HOPPing[:LIST]:ADD`

Related Commands RADar:PULSe:HOPPing[:LIST]:DELete, RADar:PULSe:HOPPing[:LIST]:SELect, RADar:PULSe:HOPPing[:LIST]:INSert

Examples `RADAR:PULSE:HOPPING:LIST:ADD` adds a single entry to the end of the active hopping pattern list for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:AOFFset

This command sets or returns the Amplitude Offset (in dB) of the currently selected hop for the selected pulse definition.

The Hopping Pattern must be set to either Custom or Pseudo Random List.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing[:LIST]:AOFFset <amp_offset>
RADar:PULSe:HOPPing[:LIST]:AOFFset?
```

Arguments <amp_offset>::= <NRf> value.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:LIST:AOFFSET -10` sets the Amplitude Offset of the currently selected hop to –10 dB for the selected pulse definition.

`RADAR:PULSE:HOPPING:LIST:AOFFSET?` might return `-15.0000000000`, indicating that the Amplitude Offset of the currently selected hop is set to –15 dB for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:COUNt? (Query Only)

This command returns the number of hops in the Hop List or the number of entries in the Frequency Avoid List for the selected pulse definition.

When the Hopping Pattern is either Custom or Pseudo Random List, the Frequency Hop List count is returned. When the Hopping Pattern is set to Pseudo Random Range, the Frequency Avoid List count is returned.

Group Hopping

Syntax `RADar:PULSe:HOPPing[:LIST]:COUNt?`

Returns A single <NR1> value.

Examples `RADAR:PULSE:HOPPING:LIST:COUNT?` might return `15`, indicating that there are 15 entries in the active hopping pattern list for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:DELete (No Query Form)

This command removes a single entry or all entries within the Frequency Hop List or the Frequency Avoid List for the selected pulse definition.

When the Hopping Pattern is either Custom or Pseudo Random List, the Frequency Hop List modified. When the Hopping Pattern is set to Pseudo Random Range, the Frequency Avoid List is modified.

Group Hopping

Syntax `RADar:PULSe:HOPPing[:LIST]:DELete {ALL|<index>}`

Related Commands RADar:PULSe:HOPPing[:LIST]:ADD, RADar:PULSe:HOPPing[:LIST]:SELect, RADar:PULSe:HOPPing[:LIST]:INSert

Arguments ALL – Deletes all entries of the current hopping list.

<index>::= <NR1> value. Specifies the index number to delete from the current hopping list.

Examples `RADAR:PULSE:HOPPING:LIST:DELETE ALL` removes all entries from the active hopping pattern list for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:FOFFset

This command sets or returns the Frequency Offset (Relative Frequency) of the currently selected hop for the selected pulse definition.

The Hopping Pattern must be set to either Custom or Pseudo Random List.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing[:LIST]:FOFFset <freq_offset>
RADar:PULSe:HOPPing[:LIST]:FOFFset?
```

Arguments <freq_offset>::= <NRf> value.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:LIST:FOFFSET 1E9` sets the Frequency Offset of the currently selected hop to 1 GHz for the selected pulse definition.

`RADAR:PULSE:HOPPING:LIST:FOFFSET?` might return `100.0000000000E+6`, indicating that the Frequency Offset of the currently selected hop is set to 100 MHz for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:FREQuency

This command sets or returns the Frequency of the currently selected hop in the Frequency Avoid List for the selected pulse definition.

The Hopping Pattern must be set to either Custom or Pseudo Random List.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing[:LIST]:FREQuency <frequency>
RADar:PULSe:HOPPing[:LIST]:FREQuency?
```

Arguments <frequency>::= <NRf> value.
Range: Minimum = 1 Hz. Maximum value is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:LIST:FREQUENCY 1E9` sets the Frequency of the currently selected hop in the Frequency Avoid List to 1 GHz for the selected pulse definition.

`RADAR:PULSE:HOPPING:LIST:FREQUENCY?` might return `100.0000000000E+6`, indicating that the Frequency of the currently selected hop in the Frequency Avoid List is set to 100 MHz for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:INSert (No Query Form)

This command inserts a single Hop to the Frequency Hop List at the specified index for the selected pulse definition.

For example, if the current Frequency Hop List contains 10 indices, inserting a new hop at index 10 creates a new hop at index 10, moving the existing hop to index 11.

Conditions The Hopping Pattern must be set to either Custom or Pseudo Random List.

The specified index must be a valid index.

Group Hopping

Syntax `RADar:PULSe:HOPPing[:LIST]:INSert <index>`

Related Commands RADar:PULSe:HOPPing[:LIST]:ADD, RADar:PULSe:HOPPing[:LIST]:DELete, RADar:PULSe:HOPPing[:LIST]:SELect

Arguments <index>::= <NR1> value.

Examples `RADAR:PULSE:HOPPING:LIST:INSERT 3` inserts a new Hop at the 3rd index, moving all previously defined indices down one row.

RADar:PULSe:HOPPing[:LIST]:PSINdex

This command sets or returns the Pulse Start Index (Start Index) of the currently selected hop for the selected pulse definition.

The Hopping Pattern must set to Custom.

The Hop Time must be set to Pulse Start Index.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing[:LIST]:PSINdex <start_index>
RADar:PULSe:HOPPing[:LIST]:PSINdex?
```

Arguments <start_index>::= <NR1> value.

Returns A single <NR1> value.

Examples `RADAR:PULSE:HOPPING:LIST:PSINDEX 10` sets the Pulse Start Index of the currently selected hop to start on Pulse index 10 for the selected pulse definition.

`RADAR:PULSE:HOPPING:LIST:PSINDEX?` might return `6`, indicating that the Pulse Start Index of the currently selected hop is set to 6 for the selected pulse definition.

RADar:PULSe:HOPPing[:LIST]:SELect

This command sets or returns the selected Hop number in the Frequency Hop List or the Index number in the Frequency Avoid List.

When the Hopping Pattern is either Custom or Pseudo Random List, the Frequency Hop List is used for selection. When the Hopping Pattern is set to Pseudo Random Range, the Frequency Avoid List is used for selection.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing[:LIST]:SELect <index>
RADar:PULSe:HOPPing[:LIST]:SELect?
```

Arguments <index>::= <NR1> value.

Returns A single <NR1> value.

Examples `RADAR:PULSE:HOPPING:LIST:SELECT 3` selects index 3 in the hopping table (Hop list or Avoid list).

`RADAR:PULSE:HOPPING:LIST:SELECT?` might return `1`, indicating that index 1 is currently selected for modification. in the active hopping pattern list (Hop list or Avoid list).

RADar:PULSe:HOPPing:PATTern

This command sets or returns the Hopping Pattern for the selected pulse definition.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:PATTern {CUSTom|LIST|RANGe}
RADar:PULSe:HOPPing:PATTern?
```

Arguments CUSTom = Hops are based on the Frequency Hop List.

LIST = Hops are chosen randomly (based on PRBS selection) from the Frequency Hop List.

RANGe = Hops are chosen randomly (based on PRBS selection) from frequencies between a minimum and maximum frequency with a minimum frequency spacing. Frequencies included in the Frequency Avoid List are skipped.

Returns CUST: Custom
LIST
RANG: Range

Examples `RADAR:PULSE:HOPPING:PATTERN LIST` sets the Hopping Pattern to choose random hops from the Frequency Hop List for the selected pulse definition.

`RADAR:PULSE:HOPPING:PATTERN?` might return `CUST`, indicating that the Hopping Pattern is based on the Frequency Hop List for the selected pulse definition.

RADar:PULSe:HOPPing:PRBS

This command sets or returns the Hopping PRBS pattern for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random List or Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:PRBS
{PRBS7|PRBS9|PRBS15|PRBS16|PRBS20|PRBS21|PRBS23|PRBS29
|PRBS31|UDEFined}
RADar:PULSe:HOPPing:PRBS?
```

Related Commands RADar:PULSe:HOPPing:PATTern,
RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial,
RADar:PULSe:HOPPing:PRBS:UDEFined:RESet

Arguments PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31, and UDEF are the PRBS types.

When setting to UDEF (User Defined), use the commands RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial and RADar:PULSe:HOPPing:PRBS:UDEFined:RESet

to define the PRBS Polynomial Expression and Shift register initial value.

Returns PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31, UDEF

Examples `RADAR:PULSE:HOPPING:PRBS PRBS21` sets the Hopping PRBS to PRBS 21 for the selected pulse definition.

`RADAR:PULSE:HOPPING:PRBS?` might return `PRBS15`, indicating that the Hopping PRBS is set to PRBS15 for the selected pulse definition.

RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial

This command sets or returns the Hopping PRBS Polynomial Expression for the User Defined PRBS for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random List or Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial <polynomial>
RADar:PULSe:HOPPing:PRBS:UDEFined:POLYnomial?
```

Related Commands RADar:PULSe:HOPPing:PATTern

Arguments <polynomial>::= <string>.

The highest degree of polynomial is 31. If the expression contains more than 31, an error will be returned. For example X32+X23+1 returns an error.

Returns A single <polynomial> string.

Examples `RADAR:PULSE:HOPPING:PRBS:UDEFINED:POLYNOMIAL "X12+X11+1"` sets the Hopping User Defined PRBS polynomial expression to X12+X11+1 for the selected pulse definition.

`RADAR:PULSE:HOPPING:PRBS:UDEFINED:POLYNOMIAL?` might return `"X12+X11+1"`, indicating this is the Hopping User Defined PRBS polynomial for the selected pulse definition.

RADar:PULSe:HOPPing:PRBS:UDEFined:RESet (No Query Form)

This command resets the Hopping User Defined PRBS Polynomial Expression and Shift register initial values to their default values.

Group Hopping

Syntax `RADar:PULSe:HOPPing:PRBS:UDEFined:RESet`

Examples `RADAR:PULSE:HOPPING:PRBS:UDEFINED:RESET` resets the Hopping User Defined PRBS values.

RADar:PULSe:HOPPing:PRBS:UDEFined:SREGister

This command sets or returns the Hopping PRBS Polynomial Expression for the User Defined PRBS for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random List or Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:PRBS:UDEFined:SREGister <polynomial>
RADar:PULSe:HOPPing:PRBS:UDEFined:SREGister?
```

Related Commands RADar:PULSe:HOPPing:PATTern

Arguments <polynomial>::= <string>.

The highest degree of polynomial is 31. If the expression contains more than 31 an error will be returned. For example X32+X23+1 returns an error.

Returns A single <polynomial> string.

Examples `RADAR:PULSE:HOPPING:PRBS:UDEFINED:SREGISTER "11011"` sets the Hopping User Defined PRBS Shift register to 11011 for the selected pulse definition.

`RADAR:PULSE:HOPPING:PRBS:UDEFINED:SREGISTER?` might return `"11011"`, indicating this is the Hopping User Defined PRBS shift register for the selected pulse definition.

RADar:PULSe:HOPPing:RANGe:ALISt[:ENABle]

This command sets or returns the Hopping Frequency Avoid List state (enabled or disabled) when the Hopping Pattern is set to Pseudo List for the selected pulse definition.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:RANGe:ALISt[:ENABle] {1|0|ON|OFF}
RADar:PULSe:HOPPing:RANGe:ALISt[:ENABle]?
```

Related Commands RADar:PULSe:HOPPing:PATTern

Arguments OFF or 0 disables Repeat List. OFF or 0 is the default value.

ON or 1 enables Repeat List.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:HOPPING:RANGE:ALIST:ENABLE 1` enables the Frequency Avoid List for the selected pulse definition.

`RADAR:PULSE:HOPPING:RANGE:ALIST:ENABLE?` might return `0`, indicating that the Frequency Avoid List is disabled for the selected pulse definition.

RADar:PULSe:HOPPing:RANGe[:FREQuency]:MAXimum

This command sets or returns the Maximum Frequency value for the Pseudo Random Range Hopping Pattern for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MAXimum <max_freq>
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MAXimum?
```

Related Commands RADar:PULSe:HOPPing:PATTern, RADar:PULSe:HOPPing:RANGe[:FREQuency]:MINimum

Arguments <max_freq>::= <NRf> value.
Range: Maximum value is instrument dependent. Minimum value can not be less than the Minimum Frequency setting.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:RANGE:FREQUENCY:MAXIMUM 2E9` sets the Pseudo Random Range Maximum Frequency to 2 GHz for the selected pulse definition.

`RADAR:PULSE:HOPPING:RANGE:FREQUENCY:MAXIMUM?` might return `2.0000000000E+9`, indicating that the Pseudo Random Range Maximum Frequency is set to 2 GHz for the selected pulse definition.

RADar:PULSe:HOPPing:RANGe[:FREQuency]:MINimum

This command sets or returns the Minimum Frequency value for the Pseudo Random Range Hopping Pattern for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MINimum <min_freq>
RADar:PULSe:HOPPing:RANGe[:FREQuency]:MINimum?
```

Related Commands RADar:PULSe:HOPPing:PATTern, RADar:PULSe:HOPPing:RANGe[:FREQuency]:MAXimum

Arguments <min_freq>::= <NRf> value.
Range: Minimum = 1 Hz, Maximum is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:RANGE:FREQUENCY:MINIMUM 1E9` sets the Pseudo Random Range Minimum Frequency to 1 GHz for the selected pulse definition.

`RADAR:PULSE:HOPPING:RANGE:FREQUENCY:MINIMUM?` might return `1.0000000000E+9`, indicating that the Pseudo Random Range Minimum Frequency is set to 1 GHz for the selected pulse definition.

RADar:PULSe:HOPPing:RANGe[:FREQuency]:SPACing

This command sets or returns the Frequency Spacing value for the Pseudo Random Range Hopping Pattern for the selected pulse definition.

The Hopping Pattern must be set to Pseudo Random Range.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:RANGe[:FREQuency]:SPACing <spacing>
RADar:PULSe:HOPPing:RANGe[:FREQuency]:SPACing?
```

Related Commands RADar:PULSe:HOPPing:PATTern

Arguments <spacing>::= <NR3> value.
Range: Minimum = 100 Hz, Maximum is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:RANGE:FREQUENCY:SPACING 1E9` sets the Pseudo Random Range Frequency Spacing to 1 MHz for the selected pulse definition.

`RADAR:PULSE:HOPPING:RANGE:FREQUENCY:SPACING?` might return `10.0000000000E+3`, indicating that the Pseudo Random Range Frequency Spacing is set to 10 kHz for the selected pulse definition.

RADar:PULSe:HOPPing:TIME

This command sets or returns the Hop Time type for the selected pulse definition.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:TIME {PPHop|PSINdex}
RADar:PULSe:HOPPing:TIME?
```

Arguments PPHop = Pulses Per Hop.

PSINdex = Pulse Start Index. (Available for Custom Hopping Pattern.)

Returns PPH
PSIN

Examples `RADAR:PULSE:HOPPING:TIME PPHop` sets the Hop Time to Pulses Per Second for the selected pulse definition.

`RADAR:PULSE:HOPPING:TIME?` might return `PPH`, indicating that the Hop Time is set to Pulses Per Second for the selected pulse definition.

RADar:PULSe:HOPPing:TIME:PPHop

This command sets or returns the Pulses per Hop for frequency hopping for the selected pulse definition.

The Hop Time must be set to Pulses Per Hop.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:TIME:PPHop <pulses>
RADar:PULSe:HOPPing:TIME:PPHop?
```

Related Commands RADar:PULSe:HOPPing:TIME

Arguments <pulses>::= <NRf> value.

Range: 0 to 500000.
Default: 0.

Returns A single <NR3> value.

Examples `RADAR:PULSE:HOPPING:TIME:PPHOP 50` sets the Pulses Per Hop to 50 pulses for the selected pulse definition.

`RADAR:PULSE:HOPPING:TIME:PPHOP?` might return `5.0000000000E+3`, indicating that the Pulses per Hop is set to 5000 pulses for the selected pulse definition.

RADar:PULSe:HOPPing:TON

This command sets or returns the frequency Hopping state (enabled or disabled) for the selected pulse definition.

Group Hopping

Syntax

```
RADar:PULSe:HOPPing:TON {0|1|OFF|ON}
RADar:PULSe:HOPPing:TON?
```

Arguments OFF or 0 disables frequency Hopping. OFF or 0 is the default value.
ON or 1 enables frequency Hopping.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:HOPPING:TON 1` enables frequency Hopping for the selected pulse definition.

`RADAR:PULSE:HOPPING:TON?` might return `0`, indicating that frequency Hopping is not enabled for the selected pulse definition.

RADar:PULSe:IMPairments:EJITter:DEViation

This command sets or returns the Edge Jitter Deviation Pulse Impairment for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:EJITter:DEViation <deviation>
RADar:PULSe:IMPairments:EJITter:DEViation?
```

Arguments <deviation>::= <NRf> value.

Range: 0 s to 100 ms.
Default: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:IMPAIRMENTS:EJITTER:DEVIATION 1E-12` sets the Edge Jitter Deviation for Pulse Impairments to 1 ps for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:EJITTER:DEVIATION?` might return `100.0000000000E-12`, indicating that the Edge Jitter Deviation for Pulse Impairments is set to 100 ps for the selected pulse definition.

RADar:PULSe:IMPairments:EJITter:DISTribution

This command sets or returns the Edge Jitter Distribution type Pulse Impairment for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:EJITter:DISTribution
{GAUSsian|UNIForm}
RADar:PULSe:IMPairments:EJITter:DISTribution?
```

Arguments

GAUSsian
UNIForm

Returns

GAUS
UNIF

Examples

`RADAR:PULSE:IMPAIRMENTS:EJITTER:DISTRIBUTION GAUSSIAN` sets Pulse Impairments Edge Jitter Distribution type to Gaussian for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:EJITTER:DISTRIBUTION?` might return `UNIF`, indicating that the Pulse Impairments Edge Jitter Distribution type is set to Uniform for the selected pulse definition.

RADar:PULSe:IMPairments:EJITter:TON

This command sets or returns the Edge Jitter Pulse Impairments state (enabled or disabled) for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:EJITter:TON {0|1|OFF|ON}
RADar:PULSe:IMPairments:EJITter:TON?
```

Arguments OFF or 0 disables Edge Jitter Pulse Impairments. OFF or 0 is the default value. ON or 1 enables Edge Jitter Pulse Impairments.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IMPAIRMENTS:EJITTER:TON 1` enables Edge Jitter Pulse Impairments for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:EJITTER:TON?` might return `0`, indicating that Edge Jitter Pulse Impairments is not enabled for the selected pulse definition.

RADar:PULSe:IMPairments:WJITter:DEViation

This command sets or returns the Width Jitter Deviation Pulse Impairment for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:WJITter:DEViation <deviation>
RADar:PULSe:IMPairments:WJITter:DEViation?
```

Arguments <deviation>::= <NRf> value.

Range: 0 s to 100 ms.
Default: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:IMPAIRMENTS:WJITTER:DEVIATION 1E-12` sets the Width Jitter Deviation for Pulse Impairments to 1 ps for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:WJITTER:DEVIATION?` might return `100.0000000000E-12`, indicating that the Width Jitter Deviation for Pulse Impairments is set to 100 ps for the selected pulse definition.

RADar:PULSe:IMPairments:WJITter:DISTribution

This command sets or returns the Width Jitter Distribution type Pulse Impairment for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:WJITter:DISTribution
{GAUSsian|UNIForm}
RADar:PULSe:IMPairments:WJITter:DISTribution?
```

Arguments

GAUSsian
UNIForm

Returns

GAUS
UNIF

Examples

`RADAR:PULSE:IMPAIRMENTS:WJITTER:DISTRIBUTION GAUSSIAN` sets Pulse Impairments Width Jitter Distribution type to Gaussian for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:WJITTER:DISTRIBUTION?` might return `UNIF`, indicating that the Pulse Impairments Width Jitter Distribution type is set to Uniform for the selected pulse definition.

RADar:PULSe:IMPairments:WJITter:TON

This command sets or returns the Width Jitter Pulse Impairments state (enabled or disabled) for the selected pulse definition.

Group Pulse impairments

Syntax

```
RADar:PULSe:IMPairments:WJITter:TON {0|1|OFF|ON}
RADar:PULSe:IMPairments:WJITter:TON?
```

Arguments OFF or 0 disables Width Jitter Pulse Impairments. OFF or 0 is the default value. ON or 1 enables Width Jitter Pulse Impairments.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IMPAIRMENTS:WJITTER:TON 1` enables Width Jitter Pulse Impairments for the selected pulse definition.

`RADAR:PULSE:IMPAIRMENTS:WJITTER:TON?` might return `0`, indicating that Width Jitter Pulse Impairments is not enabled for the selected pulse definition.

RADar:PULSe:INTerference:ANOIse:BANDwidth

This command sets or returns the additive noise Bandwidth for the selected pulse definition.

Group Interference addition

Syntax

```
RADar:PULSe:INTerference:ANOIse:BANDwidth <bandwidth>
RADar:PULSe:INTerference:ANOIse:BANDwidth?
```

Arguments <bandwidth>::= <NRf> value.
Range: Minimum = 1 Hz. Maximum is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:INTERFERENCE:ANOISE:BANDWIDTH 1E6` sets the additive noise Bandwidth to 1 MHz for the selected pulse definition.

`RADAR:PULSE:INTERFERENCE:ANOISE:BANDWIDTH?` might return `8.5000000000E+6`, indicating that the additive noise Bandwidth is set to 8.5 MHz for the selected pulse definition.

RADar:PULSe:INTerference:ANOIse:POTOnly

This command sets or returns the additive noise Pulse On Time Only state (enabled or disabled) for the selected pulse definition. When enabled, the additive noise is only applied to the on-time of the pulse.

Group Interference addition

Syntax

```
RADar:PULSe:INTerference:ANOIse:POTOnly {1|0|ON|OFF}
RADar:PULSe:INTerference:ANOIse:POTOnly?
```

Arguments OFF or 0 disables the control, and Additive Noise is applied to the entire pulse group.
OFF or 0 is the default value.

ON or 1 enables the control, applying the Additive Noise to only the pulse on-time of the pulse group.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:INTERFERENCE:ANOISE:POTONLY 1` enables applying the Additive Noise to only the pulse on-time for the selected pulse definition.

`RADAR:PULSE:INTERFERENCE:ANOISE:POTONLY?` might return `0`, indicating that Additive Noise is added to the entire pulse group for the selected pulse definition.

RADar:PULSe:INTerference:ANOIse:SNR

This command sets or returns the additive noise SNR (in dB) for the selected pulse definition.

Group Interference addition

Syntax

```
RADar:PULSe:INTerference:ANOIse:SNR <level>
RADar:PULSe:INTerference:ANOIse:SNR?
```

Arguments <level>::= <NRf> value.
Range: –80 dB to 80 dB.

Returns A single <NR2> value.

Examples `RADAR:PULSE:INTERFERENCE:ANOISE:SNR 5` sets the additive noise SNR level to 5 dB for the selected pulse definition.

`RADAR:PULSE:INTERFERENCE:ANOISE:SNR?` might return `-20.0000000000`, indicating that the additive noise SNR level is set to –20 dB for the selected pulse definition.

RADar:PULSe:INTerference:ANOIse:TON

This command sets or returns the Additive Noise interference state (enabled or disabled) for the selected pulse definition.

Group Interference addition

Syntax

```
RADar:PULSe:INTerference:ANOIse:TON {1|0|ON|OFF}
RADar:PULSe:INTerference:ANOIse:TON?
```

Arguments OFF or 0 disables Additive Noise. OFF or 0 is the default value.

ON or 1 enables Additive Noise.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:INTERFERENCE:ANOISE:TON 1` enables the Additive Noise for the selected pulse definition.

`RADAR:PULSE:INTERFERENCE:ANOISE:TON?` might return `0`, indicating that the Additive Noise is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:CLEAkage:IOFFset

This command sets or returns the carrier leakage I Offset percentage for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:CLEAkage:IOFFset <percentage>
RADar:PULSe:IQIMpairment:CLEAkage:IOFFset?
```

Related Commands RADar:PULSe:IQIMpairment:CLEAkage:QOFFset

Arguments <percentage>::= <NRf> value.
Range: –50% to 50%.

Returns A single <NR2> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:IOFFSET –10` sets the Carrier Leakage I Offset percentage to –10 % for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:IOFFSET?` might return `3.0000000000`, indicating that the Carrier Leakage I Offset percentage is set to 3 % for the selected pulse definition.

RADar:PULSe:IQIMpairment:CLEAkage:IQOFfset

This command sets or returns the carrier leakage IQ Offset (in dB) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:CLEAkage:IQOFfset <level>
RADar:PULSe:IQIMpairment:CLEAkage:IQOFfset?
```

Related Commands RADar:PULSe:IQIMpairment:CLEAkage:IOFFset, RADar:PULSe:IQIMpairment:CLEAkage:QOFFset

Arguments <level>::= <NRf> value.

Returns A single <NR2> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:IQOFFSET −5` sets the Carrier Leakage IQ Offset level to –5 dB for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:IQOFFSET?` might return `-20.0000000000`, indicating that the Carrier Leakage IQ Offset level is set to –20 dB for the selected pulse definition.

RADar:PULSe:IQIMpairment:CLEAkage:QOFFset

This command sets or returns the carrier leakage Q Offset percentage for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:CLEAkage:QOFFset <percentage>
RADar:PULSe:IQIMpairment:CLEAkage:QOFFset?
```

Related Commands RADar:PULSe:IQIMpairment:CLEAkage:IOFFset

Arguments <percentage>::= <NRf> value.
Range: –50% to 50%.

Returns A single <NR2> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:QOFFSET –10` sets the Carrier Leakage Q Offset percentage to –10 % for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:QOFFSET?` might return `3.0000000000`, indicating that the Carrier Leakage Q Offset percentage is set to 3 % for the selected pulse definition.

RADar:PULSe:IQIMpairment:CLEAkage:TON

This command sets or returns the Carrier Leakage state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:CLEAkage:TON {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:CLEAkage:TON?
```

Arguments OFF or 0 disables Carrier Leakage. OFF or 0 is the default value.

ON or 1 enables Carrier Leakage.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:TON1` enables the Carrier Leakage for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:CLEAKAGE:TON?` might return `0`, indicating that the Carrier Leakage is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:IINVert

This command sets or returns the I Invert state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:IINVert {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:IINVert?
```

Related Commands RADar:PULSe:IQIMpairment:QINVert

Arguments OFF or 0 disables inverting the I signal. OFF or 0 is the default value.

ON or 1 enables inverting the I signal.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:IINVERT 1` enables inverting the I signal for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:IINVERT?` might return `0`, indicating that inverting the I signal is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:IQIMbalance:IMBalance

This command sets or returns the Imbalance percentage for IQ Imbalance for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:IQIMbalance:IMBalance <percentage>
RADar:PULSe:IQIMpairment:IQIMbalance:IMBalance?
```

Arguments <percentage>::= <NRf> value.
Range: –30% to 30 %.

Returns A single <NR2> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:IQIMBALANCE:IMBALANCE -11` sets the IQ Imbalance percentage to –11 % for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:IQIMBALANCE:IMBALANCE?` might return `3.0000000000`, indicating that the IQ Imbalance percentage is set to 3 % for the selected pulse definition.

RADar:PULSe:IQIMpairment:IQIMbalance:TON

This command sets or returns the IQ Imbalance state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:IQIMbalance:TON {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:IQIMbalance:TON?
```

Arguments OFF or 0 disables IQ Imbalance. OFF or 0 is the default value.

ON or 1 enables IQ Imbalance.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:IQIMBALANCE:TON 1` enables the IQ Imbalance for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:IQIMBALANCE:TON?` might return `0`, indicating that the IQ Imbalance is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:IQSWap:TON

This command sets or returns the Swap I & Q state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:IQSWap:TON {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:IQSWap:TON?
```

Arguments OFF or 0 disables Swap I & Q. OFF or 0 is the default value.

ON or 1 enables Swap I & Q.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:IQSWAP:TON 1` enables Swap I & Q for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:IQSWAP:TON?` might return `0`, indicating that Swap I & Q is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:QERRor:ERRor

This command sets or returns the I/Q Error (in degrees) for Quadrature Error for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:QERRor:ERRor <degrees>
RADar:PULSe:IQIMpairment:QERRor:ERRor?
```

Arguments <degrees>::= <NRf> value.
Range: –30° to 30°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:QERROR:ERROR −11` sets the Quadrature Error I/Q Error to –11° for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:QERROR:ERROR?` might return `3.0000000000`, indicating that the Quadrature Error I/Q Error is set to 3° for the selected pulse definition.

RADar:PULSe:IQIMpairment:QERRor:TON

This command sets or returns the Quadrature Error state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:QERRor:TON {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:QERRor:TON?
```

Arguments OFF or 0 disables Quadrature Error. OFF or 0 is the default value.

ON or 1 enables Quadrature Error.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:QERROR:TON 1` enables the Quadrature Error for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:QERROR:TON?` might return `0`, indicating that the Quadrature Error is disabled for the selected pulse definition.

RADar:PULSe:IQIMpairment:QINVert

This command sets or returns the Q Invert state (enabled or disabled) for IQ Impairments for the selected pulse definition.

Group IQ impairments

Syntax

```
RADar:PULSe:IQIMpairment:QINVert {1|0|ON|OFF}
RADar:PULSe:IQIMpairment:QINVert?
```

Related Commands RADar:PULSe:IQIMpairment:IINVert

Arguments OFF or 0 disables inverting the Q signal. OFF or 0 is the default value.

ON or 1 enables inverting the Q signal.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:IQIMPAIRMENT:QINVERT 1` enables inverting the Q signal for the selected pulse definition.

`RADAR:PULSE:IQIMPAIRMENT:QINVERT?` might return `0`, indicating that inverting the Q signal is disabled for the selected pulse definition.

RADar:PULSe:MODulation

This command sets or returns the Modulation Type for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation {NMODulation|LFM|CSEQuence|UDOWn
|SFModulation|PLFM|USFMam|NLFM|FRANk|BARKer|POLYphase
|P1Polyphase|P2Polyphase|P3Polyphase|P4Polyphase|USPMam
|QPSK|BPSK|CMODulation}
```

Arguments NMODulation: No modulation

Frequency modulation types
LFM: Linear Frequency Modulation
CSEQuence: Chirp Sequence
UDOWn: Up-Down Chirp Modulation
PLFM: Piecewise LFM
SFModulation: Step Frequency Modulation
USFMam: User Defined Step FM AM
NLFM: Non Linear FM

Phase modulation types
BARKer: Biphase Coded Pulse
FRANk: Frank Code
POLYphase: Polyphase Codes
P1Polyphase: P1-Polyphase Code
P2Polyphase: P2-Polyphase Code
P3Polyphase: P3-Polyphase Code
P4Polyphase: P4-Polyphase Code
USPMam: User Defined Step PM AM

Digital modulation types
QPSK: QPSK
BPSK: BPSK

Custom modulation
CMODulation: Custom Modulation

Returns NMOD: No modulation
LFM: Linear Frequency Modulation
CSEQ: Chirp Sequence
UDOW: Up-Down Chirp Modulation
PLFM: Piecewise LFM
SFM: Step Frequency Modulation
USFM: User Defined Step FM AM

NLFM: Non Linear FM
BARK: Biphase Coded Pulse
FRAN: Frank Code
POLY: Polyphase Codes
P1P: P1-Polyphase Code
P2P: P2-Polyphase Code
P3P: P3-Polyphase Code
P4P: P4-Polyphase Code
USPM: User Defined Step PM AM
QPSK: QPSK
BPSK: BPSK
CMOD: Custom Modulation

Examples

`RADAR:PULSE:MODULATION SFMODULATION` sets the modulation Type to Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION?` might return `LFM`, indicating that the modulation Type is set to Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:BARKer:CODE

This command sets or returns the Barker code of the Barker modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:BARKer:CODE {TWO|ATWo|
THRee|ATHRee|FOUR|AFOUr|FIVe|SEVen|ELEVen|THIRteen}
RADar:PULSe:MODulation:BARKer:CODE?
```

Arguments

TWO: Two
ATWo: Alternate Two
THRee: Three
ATHRee: Alternate Three
FOUR: Four
AFOUr: Alternate Four
FIVe: Five
SEVen: Seven
ELEVen: Eleven
THIRteen: Thirteen

Returns

TWO: Two
ATW: Alternate Two
THR: Three
ATHR: Alternate Three
FOUR: Four
AFOU: Alternate Four
FIV: Five
SEV: Seven
ELEV: Eleven
THIR: Thirteen

Examples

`RADAR:PULSE:MODULATION:BARKER:CODE ATWo` sets the Barker code type to Alternate Two for the selected pulse definition.

`RADAR:PULSE:MODULATION:BARKER:CODE?` might return `THR`, indicating the Barker code type is set to Three for the selected pulse definition.

RADar:PULSe:MODulation:CSEQuence:CCOUnt

This command sets or returns the Number of Chirps for the Chirp Sequence modulation of the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:CSEQuence:CCOUnt <chirps>
RADar:PULSe:MODulation:CSEQuence:CCOUnt?
```

Arguments <chirps>::= <NR1> value.
Range: 1 to 100.
Default: 4.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:CSEQUENCE:CCOUNT 10` sets the pulse modulation Number of Chirps to 10 for the selected pulse definition.

`RADAR:PULSE:MODULATION:CSEQUENCE:CCOUNT?` might return `4.0000000000`, indicating that the pulse modulation Number of Chirps is set to 4 for the selected pulse definition.

RADar:PULSe:MODulation:CSEQuence:FSWeep

This command sets or returns the Chirp Sequence modulation Frequency Sweep method for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:CSEQuence:FSWeep {LHIGh|HLOW}`

Arguments LHIGh: Low to High.
HLOW: High to Low.

Returns LHIG: Low to High.
HLOW: High to Low.

Examples `RADAR:PULSE:MODULATION:CSEQUENCE:FSWEEP LHIGH` sets the pulse modulation Chirp Sequence Frequency Sweep method to “Low to High” for the selected pulse definition.

`RADAR:PULSE:MODULATION:CSEQUENCE:FSWEEP?` might return `HLOW`, indicating that the pulse modulation Chirp Sequence Frequency Sweep method is set to “High to Low” for the selected pulse definition.

RADar:PULSe:MODulation:CSEQuence:SRANge

This command sets or returns the Chirp Sequence modulation Sweep Range for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:CSEQuence:SRANge <sweep_range>
RADar:PULSe:MODulation:CSEQuence:SRANge?
```

Arguments <sweep_range>::= <NRf> value.
Range: 1 Hz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:CSEQUENCE:SRANGE 1E3` sets the pulse modulation Chirp Sequence Sweep Range to 1 kHz for the selected pulse definition.

`RADAR:PULSE:MODULATION:CSEQUENCE:SRANGE?` might return `10.0000000000E+6`, indicating that the pulse modulation Chirp Sequence Sweep Range is set to 10 MHz for the selected pulse definition.

RADar:PULSe:MODulation:CUSTom:PATH

This command sets or returns the filepath to custom modulation file for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:CUSTom:PATH <file_path>`

Related Commands RADar:PULSe:MODulation

Arguments <file_path>::= <string>.

Returns A single <file_path> string.

Examples `RADAR:PULSE:MODULATION:CUSTOM:PATH "c:\100usec_LFMdwn_30MHz.txt"` sets the file path to the 100usec_LFMdwn_30MHz.txt modulation file.

`RADAR:PULSE:MODULATION:CUSTOM:PATH?` might return `c:\100usec_LFMdwn_30MHz.txt`.

RADar:PULSe:MODulation:DMODulation:DATA

This command sets or returns the Data Pattern type for the selected Digital Modulation for the selected pulse definition.

When setting the pattern type to Pattern, File, or User Defined PRBS, use the additional commands noted with each argument to continue defining these pattern types. To set a PRBS pattern, see the command RADar:PULSe:MODulation:DMODulation:DATA:PRBS.

Conditions This command applies when the selected Digital Modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:DATA
{AZERo|AONE|PATTern|FILE}
```

Related Commands RADar:PULSe:MODulation:DMODulation:DATA:PRBS, RADar:PULSe:MODulation:DMODulation:DATA:PATTern, RADar:PULSe:MODulation:DMODulation:DATA:FILE

Arguments

`AZERo` is a data pattern of all zeroes.

`AONe` is a data pattern of all ones.

`PATTern` is a user defined data pattern. Use the command RADar:PULSe:MODulation:DMODulation:DATA:PATTern to set and query the pattern.

`FILe` uses a user defined file to define the data pattern. Use the command RADar:PULSe:MODulation:DMODulation:DATA:FILE to set and query the pattern file and file path.

Returns

AZER: All zeros
AON: All ones
PATT: User pattern
FIL: User file

Examples

`RADAR:PULSE:MODULATION:DMODULATION:DATA AONE` sets the modulation Data Pattern type to All ones for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:DATA?` might return `AZER`, indicating that the modulation Data Pattern type is set to All Zeros for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:DATA:FILE

This command sets or returns the filepath to the user provided data file for the Digital Modulation for the selected pulse definition.

Conditions This command applies when the digital modulation is set to QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:DATA:FILE <filepath>
RADar:PULSe:MODulation:DMODulation:DATA:FILE?
```

Related Commands RADar:PULSe:MODulation

Arguments <filepath>::= <string>.

Returns A single <filepath> string.

Examples

`RADAR:PULSE:MODULATION:DMODULATION:DATA:FILE "C:\datafile.txt"` sets the data file and path for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:DATA:FILE?` might return `"c:\datafile.txt"` for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:DATA:PATTern

This command sets or returns the user defined Data Pattern for the Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:DATA:PATTern <pattern>
RADar:PULSe:MODulation:DMODulation:DATA:PATTern?
```

Related Commands RADar:PULSe:MODulation, RADar:PULSe:MODulation:DMODulation:DATA

Arguments <pattern>::= <string>.

Returns A single <pattern> string.

Examples `RADAR:PULSE:MODULATION:DMODULATION:DATA:PATTERN"11001111"` sets the data pattern for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:DATA:PATTERN?` might return `"11001101"`, indicating this is the set data pattern for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:DATA:PRBS

This command sets or returns the PRBS Data Pattern for the Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:
DATA:PRBS {PRBS7|PRBS9|PRBS15|PRBS16
|PRBS20|PRBS21|PRBS23|PRBS29|PRBS31|UDEFined}
RADar:PULSe:MODulation:DMODulation:DATA:PRBS?
```

Arguments PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31, and UDEFined are the PRBS types.

When setting to UDEF (User Defined), use the commands RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:POLYnomial and RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:SREGister to define the PRBS Polynomial Expression and Shift register initial value.

Returns PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31, UDEF

Examples `RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS PRBS15` sets the PRBS Data Type to PRBS 15.

`RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS?` might return `PRBS15`, indicating that the PRBS Data Type is set to PRBS15.

RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:POLYnomial

This command sets or returns the PRBS Polynomial Expression for the User Defined PRBS for the Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:
POLYnomial <polynomial>
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:
POLYnomial?
```

Arguments <polynomial>::= <string>.

The highest degree of polynomial is 31. If the expression contains more than 31 an error will be returned. For example X32+X23+1 returns an error.

Returns A single <polynomial> string.

Examples `RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS:UDEFINED:POLYNOMIAL "X12+X11+1"` sets the User Defined PRBS Polynomial Expression to X12+X11+1 for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS:UDEFINED:POLYNOMIAL?` might return `"X12+X11+1"`, indicating this is the User Defined PRBS Polynomial Expression for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:RESet (No Query Form)

This command resets the User Defined PRBS polynomial expression and shift register to their default values for the selected Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:RESet`

Examples `RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS:UDEFINED:RESET` resets the User Defined PRBS polynomial expression and shift register values for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:SREGister

This command sets or returns the PRBS Shift register initial value for the User Defined PRBS for the selected Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:
SREGister <shift_register>
RADar:PULSe:MODulation:DMODulation:DATA:PRBS:UDEFined:
SREGister?
```

Arguments <shift_register>::= <string>.

The number of shift register values should match the highest polynomial degree. For example, if the user defined polynomial is "X5+X4+1", then the initial shift register should have 5 values.

Returns A single <shift_register> string.

Examples `RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS:UDEFINED:SREGISTER "11011"` sets the User Defined PRBS Shift register to 11011 for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:DATA:PRBS:UDEFINED:SREGISTER?` might return `"11011"`, indicating this is the User Defined PRBS shift register for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:FILTer:ALPHa

This command sets or returns the Alpha (filter roll off) value for the selected Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:FILTer:ALPHa <alpha>
RADar:PULSe:MODulation:DMODulation:FILTer:ALPHa?
```

Arguments <alpha>::= <NRf> value.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:DMODULATION:FILTER:ALPHA 0.23` sets the Alpha filter roll off value to 0.23 for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:ALPHA?` might return `230.0000000000E-3`, indicating that the Alpha filter roll off value is set to 0.23 for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern

This command sets or returns the Pre/Post Pattern type of the filter for the selected Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern
{WARound|ZERos|PATTern}
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern?
```

Related Commands RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST, RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE

Arguments

WARound
ZERos
PATTern

Returns

WAR: Wrap around
ZER: All zeros
PATT: Pattern

Examples

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN WAROUND` set the Pre/Post pattern type to Wrap around for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN?` might return `ZER`, indicating that the Pre/Post pattern type is set to all zeros for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST

This command sets or returns the ending pattern (Post Pattern) of the filter for the selected Digital Modulation for the selected pulse definition.

This command is only applicable when the selected Pre/Post pattern type is set to PATTern.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST <data_pattern>
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST?
```

Related Commands RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE

Arguments <data_pattern> = <string>

If the modulation is BPSK then 3 bits are considered. If the modulation is QPSK then 6 bits are considered.

Returns A single <data_pattern> string.

Examples `RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:POST "101010"` sets the ending filter pattern to 101010 for QPSK modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:POST "111"` sets the ending filter pattern to 111 for BPSK modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:POST?` might return `111000` as the QSPK post pattern for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE

This command sets or returns the starting pattern (Pre Pattern) of the filter for the selected Digital Modulation for the selected pulse definition.

This command is applicable if the selected Pre/Post pattern type is PATTern.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE <data_pattern>
RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:PRE?
```

Related Commands RADar:PULSe:MODulation:DMODulation:FILTer:PPPattern:PATTern:POST

Arguments <data_pattern> = <string>

If the modulation is BPSK then 3 bits are considered. If the modulation is QPSK then 6 bits are considered.

Returns A single <data_pattern> string.

Examples `RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:PRE "101010"` sets the start filter pattern to 101010 for QPSK modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:PRE "111"` sets the start filter pattern to 111 for BPSK modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:PPPATTERN:PATTERN:PRE?` might return `111000` as the QSPK pre pattern for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:FILTer:TYPE

This command sets or returns the Filter Type for the selected Digital Modulation for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:FILTer:TYPE
{RECTangular|RCOSine|RRCosine}
RADar:PULSe:MODulation:DMODulation:FILTer:TYPE?
```

Arguments

RECTangular
RCOSine
RRCosine

Returns

RECT: Rectangular
RCOS: Raised Cosine
RRC: Root Raised Cosine

Examples `RADAR:PULSE:MODULATION:DMODULATION:FILTER:TYPE RCOSINE` sets the Filter Type to Root Raised Cosine for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:FILTER:TYPE?` might return `RRC`, indicating that the Filter Type is set to Root Raised Cosine for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:SSYMbols

This command sets or returns the state (enabled or disabled) to use the same symbols across all repeated pulses for the selected pulse definition.

Conditions This command applies when the selected digital modulation is QPSK or BPSK. This can only be enabled if the repeat count for pulse group is more than 1. The repeat count can be set using the command RADar:PULSe:REPeat.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:DMODulation:SSYMbols {1|0|ON|OFF}`

Related Commands RADar:PULSe:REPeat, RADar:PULSe:MODulation:DMODulation:SYMBols

Arguments OFF or 0 disables using the same symbols across all pulses. OFF or 0 is the default value.

ON or 1 enables using the same symbols across all pulses.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:MODULATION:DMODULATION:SSYMBOLS 1` enables repeating the set number of symbols for BPSK or QPSK modulation types across all repeated pulses for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:SSYMBOLS?` might return `0`, indicating that the set number of symbols for BPSK or QPSK modulation types will not be repeated across all pulses for the selected pulse definition.

RADar:PULSe:MODulation:DMODulation:SYMBols

This command sets or returns the number of symbols used while applying the selected Digital Modulation on the pulse for the selected pulse definition.

This command is applicable if the selected modulation is QPSK or BPSK.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:DMODulation:SYMBols <symbols>
RADar:PULSe:MODulation:DMODulation:SYMBols?
```

Related Commands RADar:PULSe:MODulation:DMODulation:SSYMbols

Arguments <symbols>::= <NR3> value.
Range: 1 to 1000.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:DMODULATION:SYMBOLS 12` sets the Digital Modulation Number of Symbols to 12 for the selected for the selected pulse definition.

`RADAR:PULSE:MODULATION:DMODULATION:SYMBOLS?` might return `12.0000000000`, indicating that the selected Digital Modulation Symbol count is set to 12 for the selected pulse definition.

RADar:PULSe:MODulation:FRANk:CLENgth

This command sets or returns the Frank code length when the Modulation Phase is set to Frank Code for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:FRANk:CLENgth <code_length>
RADar:PULSe:MODulation:FRANk:CLENgth?
```

Related Commands RADar:PULSe:MODulation

Arguments <code_length> = <NR1> value.

Range: 1 to 10

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:FRANK:CLENGTH 2` sets the pulse modulation Frank code length to 2 for the selected pulse definition.

`RADAR:PULSE:MODULATION:FRANK:CLENGTH?` might return `1`, indicating that the pulse modulation Frank code length is set to 1 for the selected pulse definition.

RADar:PULSe:MODulation:LFM:FSWeep

This command sets or returns the Linear Frequency Modulation (LFM) Frequency Sweep method for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:LFM:FSWeep {LHIGh|HLOW}`

Arguments LHIGh: Low to High.
HLOW: High to Low.

Returns LHIG: Low to High.
HLOW: High to Low.

Examples `RADAR:PULSE:MODULATION:LFM:FSWEEP LHIGH` sets the pulse modulation Linear Frequency Modulation Frequency Sweep method to “Low to High” for the selected pulse definition.

`RADAR:PULSE:MODULATION:LFM:FSWEEP?` might return `HLOW`, indicating that the pulse modulation Linear Frequency Modulation Frequency Sweep method is set to “High to Low” for the selected pulse definition.

RADar:PULSe:MODulation:LFM:SRANge

This command sets or returns the Linear Frequency Modulation (LFM) Sweep Range for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:LFM:SRANge <sweep_range>
RADar:PULSe:MODulation:LFM:SRANge?
```

Arguments <sweep_range>::= <NRf> value.
Range: 1 Hz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:LFM:SRANGE 1E3` sets the pulse modulation Linear Frequency Modulation (LFM) Sweep Range to 1 kHz for the selected pulse definition.

`RADAR:PULSE:MODULATION:LFM:SRANGE?` might return `10.0000000000E+6`, indicating that the pulse modulation Linear Frequency Modulation (LFM) Sweep Range is set to 10 MHz for the selected pulse definition.

RADar:PULSe:MODulation:NLFM:BANDwidth

This command sets or returns the Bandwidth for the Non Linear FM modulation for the selected pulse definition.

Conditions The Non Linear FM modulation profile must be either Taylor Weighted Symmetrical or Taylor Weighted Non Symmetrical.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:NLFM:BANDwidth <bandwidth>
RADar:PULSe:MODulation:NLFM:BANDwidth?
```

Arguments <bandwidth>::= <NRf> value.
Range: 1 Hz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:NLFM:BANDWIDTH 1E3` sets the Bandwidth to 1 kHz for the Non Linear FM modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:NLFM:BANDWIDTH?` might return `10.0000000000E+6`, indicating that the Bandwidth is set to 10 MHz for the Non Linear FM modulation for the selected pulse definition.

RADar:PULSe:MODulation:NLFM:PROFile

This command sets or returns the Profile for the Non Linear Frequency modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:NLFM:PROFile
{TWSYmmetrical|TWNSymmetrical|UDCoefficients}
```

Arguments

TWSYmmetrical: Taylor Weighted Symmetrical
TWNSymmetrical: Taylor Weighted Non Symmetrical
UDCoefficients: User Defined Coefficients

Returns

TWSY
TWNS
UDC

Examples

`RADAR:PULSE:MODULATION:NLFM:PROFILE TWSYMMETRICAL` sets the Non Linear FM modulation profile to Taylor Weighted Symmetrical for the selected pulse definition.

`RADAR:PULSE:MODULATION:NLFM:PROFILE?` might return `TWNS`, indicating that the Non Linear FM modulation profile is set to Taylor Weighted Non Symmetrical for the selected pulse definition.

RADar:PULSe:MODulation:NLFM:UDCPath

This command sets or returns the filepath to the User Defined Coefficients profile for the Non Linear FM modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:NLFM:UDCPath <filepath>
RADar:PULSe:MODulation:NLFM:UDCPath?
```

Arguments <filepath>:=<string> defines the path to the custom pulse shape file.

Returns A single <filepath> string.

Examples `RADAR:PULSE:MODULATION:NLFM:UDCPATH` "C:\Coefficients.txt" sets the user defined Non Linear FM modulation profile filename and filepath for the selected pulse definition.

`RADAR:PULSE:MODULATION:NLFM:UDCPATH?` might return "C:\Coefficients.txt".

RADar:PULSe:MODulation:P1Polyphase:CLENgth

This command sets or returns the P1 Polyphase code length when the Modulation Phase is set to P1 Polyphase code for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:P1Polyphase:CLENgth <code_length>
RADar:PULSe:MODulation:P1Polyphase:CLENgth?
```

Related Commands RADar:PULSe:MODulation

Arguments <code_length> = <NR1> value.

Range: 1 to 10

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:P1POLYPHASE:CLENGTH 2` sets the pulse modulation P1 Polyphase code length to 2 for the selected pulse definition.

`RADAR:PULSE:MODULATION:P1POLYPHASE:CLENGTH?` might return `1`, indicating that the pulse modulation P1 Polyphase code length is set to 1 for the selected pulse definition.

RADar:PULSe:MODulation:P2Polyphase:CLENgth

This command sets or returns the P2 Polyphase code length when the Modulation Phase is set to P2 Polyphase code for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:P2Polyphase:CLENgth <code_length>
RADar:PULSe:MODulation:P2Polyphase:CLENgth?
```

Related Commands RADar:PULSe:MODulation

Arguments <code_length> = <NR1> value.

Range: 1 to 10

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:P2POLYPHASE:CLENGTH 2` sets the pulse modulation P2 Polyphase code length to 2 for the selected pulse definition.

`RADAR:PULSE:MODULATION:P2POLYPHASE:CLENGTH?` might return `1`, indicating that the pulse modulation P2 Polyphase code length is set to 1 for the selected pulse definition.

RADar:PULSe:MODulation:P3Polyphase:CLENgth

This command sets or returns the P3 Polyphase code length when the Modulation Phase is set to P3 Polyphase code for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:P3Polyphase:CLENgth <code_length>
RADar:PULSe:MODulation:P3Polyphase:CLENgth?
```

Related Commands RADar:PULSe:MODulation

Arguments <code_length> = <NR1> value.

Range: 1 to 10

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:P3POLYPHASE:CLENGTH 2` sets the pulse modulation P3 Polyphase code length to 2 for the selected pulse definition.

`RADAR:PULSE:MODULATION:P3POLYPHASE:CLENGTH?` might return `1`, indicating that the pulse modulation P3 Polyphase code length is set to 1 for the selected pulse definition.

RADar:PULSe:MODulation:P4Polyphase:CLENgth

This command sets or returns the P4 Polyphase code length when the Modulation Phase is set to P4 Polyphase code for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:P4Polyphase:CLENgth <code_length>
RADar:PULSe:MODulation:P4Polyphase:CLENgth?
```

Related Commands RADar:PULSe:MODulation

Arguments <code_length> = <NR1> value.

Range: 1 to 10

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:P4POLYPHASE:CLENGTH 2` sets the pulse modulation P4 Polyphase code length to 2 for the selected pulse definition.

`RADAR:PULSE:MODULATION:P4POLYPHASE:CLENGTH?` might return `1`, indicating that the pulse modulation P4 Polyphase code length is set to 1 for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:ADD (No Query Form)

This command adds the specified number of steps to the Piecewise Linear Frequency Modulation table for the selected pulse definition.

The total number of steps allowed in the Piecewise table is 10 steps. The duration of each step is set such that the total duration time of all steps equals the pulse on time.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:PLFM:ADD <steps>`

Arguments <steps>::= <NR1> value.

Examples `RADAR:PULSE:MODULATION:PLFM:ADD 3` adds three additional steps to the end of the existing steps in the Piecewise table for the Piecewise Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:IOFFset

This command sets or returns the Initial Offset (frequency) for the Piecewise Linear Frequency Modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:PLFM:IOFFset <initial_offset>
RADar:PULSe:MODulation:PLFM:IOFFset?
```

Arguments <initial_offset>::= <NRf> value.
Range: –25 GHz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:PLFM:IOFFSET 1E9` sets the Initial Offset frequency to 1 GHz for the Piecewise Linear Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:PLFM:IOFFSET?` might return `10.0000000000E+6`, indicating that the Initial Offset frequency is set to 10 MHz for the Piecewise Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:STEP[n]:DELete (No Query Form)

This command deletes the specified step from the Piecewise Linear Frequency Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:PLFM:STEP[n]:DELete`

Arguments [n]::= <NR1> (“n” determines the step index number in the Piecewise table).

If omitted, n is interpreted as 1.

Examples `RADAR:PULSE:MODULATION:PLFM:STEP3:DELETE` deletes step 3 from the Piecewise Linear Frequency Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:STEP[n]:DURation

This command sets or returns the Duration of the specified step within the Piecewise Linear Frequency Modulation table for the selected pulse definition.

Conditions The duration sum of all steps within the Piecewise table cannot exceed the pulse's on time.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:PLFM:STEP[n]:DURation <duration>
RADar:PULSe:MODulation:PLFM:STEP[n]:DURation?
```

Arguments [n]::= <NR1> ("n" determines the step index number in the Piecewise table).

If omitted, n is interpreted as 1.

<duration>::= <NRf> value.
Range: The minimum value is 0 s. The maximum value is dependent on the pulse's on time and existing step durations.
Default: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:PLFM:STEP2:DURATION 10E-9` sets the Duration of step 2 to 10 ns for the Piecewise Linear Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:PLFM:STEP2:DURATION?` might return `400.0000000000E-9`, indicating that the Duration of step 2 is set to 400 ns for the Piecewise Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:STEP[n]:FSWeep

This command sets or returns the Frequency Sweep type of the specified step within the Piecewise Linear Frequency Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:PLFM:STEP[n]:FSWeep {LHIGh|HLOW}
RADar:PULSe:MODulation:PLFM:STEP[n]:FSWeep?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the Piecewise table).

If omitted, n is interpreted as 1.

LHIGh: Low to High.
HLOW: High to Low.
Default: LHIGh

Returns LHIG: Low to High.
HLOW: High to Low.

Examples `RADAR:PULSE:MODULATION:PLFM:STEP2:FSWEEP HLOW` sets the Frequency Sweep type of step 2 to High-to-low for the Piecewise Linear Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:PLFM:STEP2:FSWEEP?` might return `LHIG`, indicating that the Frequency Sweep type of step 2 is set to Low to High for the Piecewise Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:PLFM:STEP[n]:SRANge

This command sets or returns the Sweep Range of the specified step within the Piecewise Linear Frequency Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:PLFM:STEP[n]:SRANge <sweep_range>
RADar:PULSe:MODulation:PLFM:STEP[n]:SRANge?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the Piecewise table).

If omitted, n is interpreted as 1.

<sweep_range>::= <NRf> value.
Range: 100 Hz to 25 GHz
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:PLFM:STEP2:SRANGE 100E6` sets the Sweep Range of step 2 to 100 MHz for the Piecewise Linear Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:PLFM:STEP2:SRANGE?` might return `1.0000000000E+9`, indicating that the Sweep Range of step 2 is set to 1 GHz for the Piecewise Linear Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:POLYphase:IOFFset

This command sets or returns the Initial Offset (initial phase offset) for the Polyphase Codes modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:POLYphase:IOFFset <initial_offset>
RADar:PULSe:MODulation:POLYphase:IOFFset?
```

Arguments <initial_offset>::= <NR3> value.

Range: –180° to 180°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:POLYPHASE:IOFFSET –10` sets the Initial Phase Offset to –10° for the Polyphase Codes modulation type for the selected pulse definition.

`RADAR:PULSE:MODULATION:POLYPHASE:IOFFSET?` might return `50.0000000000`, indicating that the Initial Phase Offset is set to 50° for the Polyphase Codes modulation type for the selected pulse definition.

RADar:PULSe:MODulation:POLYphase:POFFset

This command sets or returns the Phase Offset for the Polyphase Codes modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:POLYphase:POFFset <phase_offset>
RADar:PULSe:MODulation:POLYphase:POFFset?
```

Arguments <phase_offset>::= <NR3> value.

Range: –180° to 180°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:POLYPHASE:POFFSET –10` sets the Phase Offset to –10° for the Polyphase Codes modulation type for the selected pulse definition.

`RADAR:PULSE:MODULATION:POLYPHASE:POFFSET?` might return `50.0000000000`, indicating that the Phase Offset is set to 50° for the Polyphase Codes modulation type for the selected pulse definition.

RADar:PULSe:MODulation:POLYphase:RAMP

This command sets or returns the Ramp Function for the Polyphase Codes modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:POLYphase:RAMP {LINear|COSine}
RADar:PULSe:MODulation:POLYphase:RAMP
```

Arguments

LINear
COSine

Returns

LIN: Linear
COS: Cosine

Examples

`RADAR:PULSE:MODULATION:POLYPHASE:RAMP LINEAR` sets the Ramp Function to Linear for the Polyphase Codes modulation type for the selected pulse definition.

`RADAR:PULSE:MODULATION:POLYPHASE:RAMP?` might return `COS`, indicating that the Ramp Function is set to Cosine for the Polyphase Codes modulation type for the selected pulse definition.

RADar:PULSe:MODulation:POLYphase:RDURation

This command sets or returns the Ramp Duration for the Polyphase Codes modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:POLYphase:RDURation <ramp_duration>
RADar:PULSe:MODulation:POLYphase:RDURation?
```

Arguments <ramp_duration>::= <NR3> value.

Range: 0% to 100%

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:POLYPHASE:RDURATION 10` sets the Ramp Duration to 10% for the Polyphase Codes modulation type for the selected pulse definition.

`RADAR:PULSE:MODULATION:POLYPHASE:RDURATION?` might return `50.0000000000`, indicating that the Ramp Duration is set to 50% for the Polyphase Codes modulation type for the selected pulse definition.

RADar:PULSe:MODulation:POLYphase:STEPs

This command sets or returns the Step Count for the Polyphase Codes modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:POLYphase:STEPs <steps>
RADar:PULSe:MODulation:POLYphase:STEPs?
```

Arguments <steps>::= <NR3> value.

Range: 1 to 200.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:POLYPHASE:STEPS 10` sets the Step Count to 10 for the Polyphase Codes modulation type for the selected pulse definition.

`RADAR:PULSE:MODULATION:POLYPHASE:STEPS?` might return `50.0000000000`, indicating that the Step Count is set to 50 for the Polyphase Codes modulation type for the selected pulse definition.

RADar:PULSe:MODulation:SFM:FOFFset

This command sets or returns the Frequency Offset for the Step Frequency Modulation (SFM) for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:SFM:FOFFset <freq_offset>
RADar:PULSe:MODulation:SFM:FOFFset?
```

Arguments <freq_offset>::= <NRf> value.
Range: –25 GHz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:SFM:FOFFSET 1E9` sets the Frequency Offset to 1 GHz for the Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:SFM:FOFFSET?` might return `10.0000000000E+6`, indicating that the Frequency Offset is set to 10 MHz for the Step Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:SFM:IOFFset

This command sets or returns the Initial Frequency Offset for the Step Frequency Modulation (SFM) for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:SFM:IOFFset <initial_offset>
RADar:PULSe:MODulation:SFM:IOFFset?
```

Arguments <initial_offset>::= <NRf> value.
Range: –25 GHz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:SFM:IOFFSET 1E9` sets the Initial Frequency Offset to 1 GHz for the Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:SFM:IOFFSET?` might return `10.0000000000E+6`, indicating that the Initial Frequency Offset is set to 10 MHz for the Step Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:SFM:RAMP

This command sets or returns the Ramp Function type for the Step Frequency Modulation (SFM) for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:SFM:RAMP {LINear|COSine}
RADar:PULSe:MODulation:SFM:RAMP?
```

Arguments

LINear
COSine

Returns

LIN: Linear
COS: Cosine

Examples

`RADAR:PULSE:MODULATION:SFM:RAMP LINEAR` sets the Ramp Function type to Linear for the Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:SFM:RAMP?` might return `COS`, indicating that the Ramp Function type is set to Cosine for the Step Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:SFM:RDURation

This command sets or returns the Ramp Duration for the Step Frequency Modulation (SFM) for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:SFM:RDURation <ramp_duration>
RADar:PULSe:MODulation:SFM:RDURation?
```

Arguments <ramp_duration>::= <NRf> value.
Range: 0% to 100%.
Default: 0%.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:SFM:RDURATION 20` sets the Ramp Duration to 20% for the Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:SFM:RDURATION?` might return `10.0000000000`, indicating that the Ramp Duration is set to 10% for the Step Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:SFM:STEPs

This command sets or returns the number of steps (Step Count) for the Step Frequency Modulation (SFM) for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:SFM:STEPs <steps>
RADar:PULSe:MODulation:SFM:STEPs?
```

Arguments <steps>::= <NRf> value.
Range: 1 to 200.
Default: 4.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:SFM:STEPS 10` sets the Step Count to 10 for the Step Frequency Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:SFM:STEPS?` might return `4.0000000000`, indicating that the Step Count is set to 4 for the Step Frequency Modulation for the selected pulse definition.

RADar:PULSe:MODulation:UDCHirp:COUNt

This command sets or returns the Up-Down Count for the Up-Down Chirp modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:UDCHirp:COUNt <count>
RADar:PULSe:MODulation:UDCHirp:COUNt?
```

Arguments <count>::= <NR1> value.
Range: 1 to 12.
Default: 1.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:UDCHIRP:COUNT 10` sets the Count to 10 for the Up-Down Chirp modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:UDCHIRP:COUNT?` might return `4.0000000000`, indicating that the Count is set to 4 for the Up-Down Chirp modulation for the selected pulse definition.

RADar:PULSe:MODulation:UDCHirp:INVert

This command sets or returns Invert state (enabled or disabled) of the Up-Down Chirp modulation for the selected pulse definition. When enabled, the frequency profile is inverted.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:UDCHirp:INVert {1|0|ON|OFF}
RADar:PULSe:MODulation:UDCHirp:INVert?
```

Arguments OFF or 0 disables the Invert state. OFF or 0 is the default value.
ON or 1 enables the Invert state.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:MODULATION:UDCHIRP:INVERT 1` inverts the frequency profile for the Up-Down Chirp modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:UDCHIRP:INVERT?` might return `0`, indicating that the frequency profile for the Up-Down Chirp modulation is not inverted for the selected pulse definition.

RADar:PULSe:MODulation:UDCHirp:SRANge

This command sets or returns the Sweep Range for the Up-Down Chirp modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:UDCHirp:SRANge <sweep_range>
RADar:PULSe:MODulation:UDCHirp:SRANge?
```

Arguments <sweep_range>::= <NRf> value.
Range: 1 Hz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:UDCHIRP:SRANGE 1E3` sets the Sweep Range to 1 kHz for the Up-Down Chirp modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:UDCHIRP:SRANGE?` might return `10.0000000000E+6`, indicating that the Sweep Range is set to 10 MHz for the Up-Down Chirp modulation for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:ADD (No Query Form)

This command adds the specified number of steps to the User Defined Step FM AM Modulation table for the selected pulse definition.

The total number of steps allowed in the User Defined Step FM AM table is 10 steps. The duration of each step is set such that the total duration time of all steps equals the pulse on time.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:USFMam:ADD <steps>`

Arguments <steps>::= <NR1> value.

Examples `RADAR:PULSE:MODULATION:USFMAM:ADD 3` adds three additional steps to the end of the existing steps in the User Defined Step FM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:RAMP

This command sets or returns the Ramp Function type for the User Defined Step FM AM Modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USFMam:RAMP {LINear|COSine}
RADar:PULSe:MODulation:USFMam:RAMP?
```

Arguments

LINear
COSine

Returns

LIN: Linear
COS: Cosine

Examples

`RADAR:PULSE:MODULATION:USFMAM:RAMP LINEAR` sets the Ramp Function type to Linear for the User Defined Step FM AM Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:USFMAM:RAMP?` might return `COS`, indicating that the Ramp Function type is set to Cosine for the User Defined Step FM AM Modulation for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:RDURation

This command sets or returns the Ramp Duration for the User Defined Step FM AM Modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USFMam:RDURation <ramp_duration>
RADar:PULSe:MODulation:USFMam:RDURation?
```

Arguments <ramp_duration>::= <NRf> value.
Range: 0% to 100%.
Default: 0%.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:USFMAM:RDURATION 20` sets the Ramp Duration to 20% for the User Defined Step FM AM Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:USFMAM:RDURATION?` might return `10.0000000000`, indicating that the Ramp Duration is set to 10% for the User Defined Step FM AM Modulation for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:STEP[n]:AMPLitude

This command sets or returns the Amplitude of the specified step for the User Defined Step FM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USFMam:STEP[n]:AMPLitude <amplitude>
RADar:PULSe:MODulation:USFMam:STEP[n]:AMPLitude?
```

Arguments [n]::= <NR1> ("n" determines the step index number in the User Defined Step FM AM table).

If omitted, n is interpreted as 1.

<amplitude>::= <NRf> value.
Range: –100 dB to 0 dB.
Default: 0 dB.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:USFMAM:STEP2:AMPLITUDE –10` sets the Amplitude of step 2 to –10 dB for the User Defined Step FM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USFMAM:STEP2:AMPLITUDE?` might return `-6.0000000000`, indicating that the Amplitude of step 2 is set to –6 dB for the User Defined Step FM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:STEP[n]:DELete (No Query Form)

This command deletes the specified step from the User Defined Step FM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:USFMam:STEP[n]:DELete`

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step FM AM Modulation table).

If omitted, n is interpreted as 1.

Examples `RADAR:PULSE:MODULATION:USFMAM:STEP3:DELETE` deletes step 3 from the User Defined Step FM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:STEP[n]:DURation

This command sets or returns the Duration of the specified step within the User Defined Step FM AM Modulation table for the selected pulse definition.

Conditions The duration sum of all steps within the User Defined Step FM AM table cannot exceed the pulse's on time.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USFMam:STEP[n]:DURation <duration>
RADar:PULSe:MODulation:USFMam:STEP[n]:DURation?
```

Arguments [n]::= <NR1> ("n" determines the step index number in the User Defined Step FM AM table).

If omitted, n is interpreted as 1.

<duration>::= <NRf> value.
Range and default values are dependent on the pulse's on time and step durations.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:USFMAM:STEP2:DURATION 10E-9` sets the Duration of step 2 to 10 ns for the User Defined Step FM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USFMAM:STEP2:DURATION?` might return `400.0000000000E-9`, indicating that the Duration of step 2 is set to 400 ns for the User Defined Step FM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USFMam:STEP[n]:FOFFset

This command sets or returns the Frequency Offset of the specified step for the User Defined Step FM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USFMam:STEP[n]:FOFFset <freq_offset>
RADar:PULSe:MODulation:USFMam:STEP[n]:FOFFset?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step FM AM table).

If omitted, n is interpreted as 1.

<freq_offset>::= <NRf> value.
Range: –25 GHz to 25 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:USFMAM:STEP2:FOFFSET 1E9` sets the Frequency Offset of step 2 to 1 GHz for the User Defined Step FM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USFMAM:STEP2:FOFFSET?` might return `10.0000000000E+6`, indicating that the Frequency Offset of step 2 is set to 10 MHz for the User Defined Step FM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:ADD (No Query Form)

This command adds the specified number of steps to the User Defined Step PM AM Modulation table for the selected pulse definition.

The total number of steps allowed in the User Defined Step PM AM table is 10 steps. The duration of each step is set such that the total duration time of all steps equals the pulse on time.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:USPMam:ADD <steps>`

Arguments <steps>::= <NR1> value.

Examples `RADAR:PULSE:MODULATION:USPMAM:ADD 3` adds three additional steps to the end of the existing steps in the User Defined Step PM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:RAMP

This command sets or returns the Ramp Function type for the User Defined Step PM AM Modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USPMam:RAMP {LINear|COSine}
RADar:PULSe:MODulation:USPMam:RAMP?
```

Arguments

LINear
COSine

Returns

LIN: Linear
COS: Cosine

Examples

`RADAR:PULSE:MODULATION:USPMAM:RAMP LINEAR` sets the Ramp Function type to Linear for the User Defined Step PM AM Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:USPMAM:RAMP?` might return `COS`, indicating that the Ramp Function type is set to Cosine for the User Defined Step PM AM Modulation for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:RDURation

This command sets or returns the Ramp Duration for the User Defined Step PM AM Modulation for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USPMam:RDURation <ramp_duration>
RADar:PULSe:MODulation:USPMam:RDURation?
```

Arguments <ramp_duration>::= <NRf> value.
Range: 0% to 100%.
Default: 0%.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:USPMAM:RDURATION 20` sets the Ramp Duration to 20% for the User Defined Step PM AM Modulation for the selected pulse definition.

`RADAR:PULSE:MODULATION:USPMAM:RDURATION?` might return `10.0000000000`, indicating that the Ramp Duration is set to 10% for the User Defined Step PM AM Modulation for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:STEP[n]:AMPLitude

This command sets or returns the Amplitude of the specified step for the User Defined Step PM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USPMam:STEP[n]:AMPLitude <amplitude>
RADar:PULSe:MODulation:USPMam:STEP[n]:AMPLitude?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step FM AM table).

If omitted, n is interpreted as 1.

<amplitude>::= <NRf> value.
Range: –100 dB to 0 dB.
Default: 0 dB.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MODULATION:USPMAM:STEP2:AMPLITUDE –10` sets the Amplitude of step 2 to –10 dB for the User Defined Step PM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USPMAM:STEP2:AMPLITUDE?` might return `-6.0000000000`, indicating that the Amplitude of step 2 is set to –6 dB for the User Defined Step PM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:STEP[n]:DELete (No Query Form)

This command deletes the specified step from the User Defined Step PM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax `RADar:PULSe:MODulation:USPMam:STEP[n]:DELete`

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step FM AM Modulation table).

If omitted, n is interpreted as 1.

Examples `RADAR:PULSE:MODULATION:USPMAM:STEP3:DELETE` deletes step 3 from the User Defined Step PM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:STEP[n]:DURation

This command sets or returns the Duration of the specified step within the User Defined Step PM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USPMam:STEP[n]:DURation <duration>
RADar:PULSe:MODulation:USPMam:STEP[n]:DURation?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step FM AM table).

If omitted, n is interpreted as 1.

<duration>::= <NRf> value.
Range and default values are dependent on the pulse’s on time and step durations.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:USPMAM:STEP2:DURATION 10E-9` sets the Duration of step 2 to 10 ns for the User Defined Step PM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USPMAM:STEP2:DURATION?` might return `400.0000000000E-9`, indicating that the Duration of step 2 is set to 400 ns for the User Defined Step PM AM Modulation table for the selected pulse definition.

RADar:PULSe:MODulation:USPMam:STEP[n]:POFFset

This command sets or returns the Phase Offset of the specified step for the User Defined Step PM AM Modulation table for the selected pulse definition.

Group Pulse modulation

Syntax

```
RADar:PULSe:MODulation:USPMam:STEP[n]:POFFset <phase_offset>
RADar:PULSe:MODulation:USPMam:STEP[n]:POFFset?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the User Defined Step PM AM table).

If omitted, n is interpreted as 1.

<phase_offset>::= <NR3> value.

Range: –180° to 180°.
Default: 0°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MODULATION:USPMAM:STEP2:FOFFSET 10` sets the Phase Offset of step 2 to 10° for the User Defined Step PM AM Modulation table for the selected pulse definition.

`RADAR:PULSE:MODULATION:USPMAM:STEP2:FOFFSET?` might return `10.0000000000`, indicating that the Phase Offset of step 2 is set to 10° for the User Defined Step PM AM Modulation table for the selected pulse definition.

RADar:PULSe:MULTipath:ADD (No Query Form)

This command adds the specified number of paths in the Multipath table for the selected pulse definition.

Group Multipath

Syntax `RADar:PULSe:MULTipath:ADD <paths>`

Related Commands RADar:PULSe:MULTipath:DELay, RADar:PULSe:MULTipath:AMPLitude, RADar:PULSe:MULTipath:PHASe

Arguments <paths>::= <NR1> value.

Examples `RADAR:PULSE:MULTIPATH:ADD 2` adds 2 new paths to the end of the Multipath table for the selected pulse definition.

RADar:PULSe:MULTipath:AMPLitude

This command sets or returns the Multipath Amplitude of the currently selected path in the multipath table for the selected pulse definition.

Group Multipath

Syntax

```
RADar:PULSe:MULTipath:AMPLitude <amplitude>
RADar:PULSe:MULTipath:AMPLitude?
```

Related Commands RADar:PULSe:MULTipath:SELect, RADar:PULSe:MULTipath:DELay, RADar:PULSe:MULTipath:PHASe

Arguments <amplitude>::= <NRf> value.
Range: –100 dB to 0 dB.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MULTIPATH:AMPLITUDE –10` sets the Multipath Amplitude of the currently selected path to –10 dB for the selected pulse definition.

`RADAR:PULSE:MULTIPATH:AMPLITUDE?` might return `-200.0000000000E-3`, indicating that the Multipath Amplitude of the currently selected path is set to –0.2 dB. for the selected pulse definition.

RADar:PULSe:MULTipath:DELay

This command sets or returns the Multipath Delay in seconds of the currently selected path in the multipath table for the selected pulse definition.

Group Multipath

Syntax

```
RADar:PULSe:MULTipath:DELay <delay>
RADar:PULSe:MULTipath:DELay?
```

Related Commands RADar:PULSe:MULTipath:SELect, RADar:PULSe:MULTipath:AMPLitude, RADar:PULSe:MULTipath:PHASe

Arguments <delay>::= <NRf> value.
Range: –1 s to 1 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MULTIPATH:DELAY 10E-3` sets the Multipath Delay to 100 ms for the selected pulse definition.

`RADAR:PULSE:MULTIPATH:DELAY?` might return `3.0000000000E-6`, indicating that the Multipath Delay is set to 3 µs for the selected pulse definition.

RADar:PULSe:MULTipath:PHASe

This command sets or returns the Multipath Phase of the currently selected path in the multipath table for the selected pulse definition.

Group Multipath

Syntax

```
RADar:PULSe:MULTipath:PHASe <phase>
RADar:PULSe:MULTipath:PHASe?
```

Related Commands RADar:PULSe:MULTipath:SELect, RADar:PULSe:MULTipath:AMPLitude, RADar:PULSe:MULTipath:DELay

Arguments <phase>::= <NRf> value.
Range: –180° to 180°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:MULTIPATH:PHASE 10` sets the Multipath Phase of the currently selected path to 10 degrees of the selected pulse definition.

`RADAR:PULSE:MULTIPATH:PHASE?` might return `-20.0000000000`, indicating that the Multipath Phase of the currently selected path is set to –20 degrees of the selected pulse definition.

RADar:PULSe:MULTipath:REMove (No Query Form)

This command deletes the specified index (row) in the Multipath table for the selected pulse definition.

If the path doesn't exist, an error is returned.

Group Multipath

Syntax `RADar:PULSe:MULTipath:REMove <index>`

Arguments <index>::= <NR1> value.

Examples `RADAR:PULSE:MULTIPATH:REMOVE 3` deletes row 3 in the Multipath table for the selected pulse definition.

RADar:PULSe:MULTipath:SELect

This command sets or returns the selected MultiPath table index (row) for the selected pulse definition.

Other MultiPath commands act on the selected table index.

Group Multipath

Syntax

```
RADar:PULSe:MULTipath:SELect <index>
RADar:PULSe:MULTipath:SELect?
```

Arguments <index>::= <NR1> value.

Returns A single <NR1> value.

Examples `RADAR:PULSE:MULTIPATH:SELECT 1` selects the first path in the MultiPath table for the selected pulse definition.

`RADAR:PULSE:MULTIPATH:SELECT?` might return `2.0000000000`, indicating that the second path in the MultiPath table is selected for the selected pulse definition.

RADar:PULSe:MULTipath:TON

This command sets or returns the Multipath state (enabled or disabled) for the selected pulse definition.

Group Multipath

Syntax

```
RADar:PULSe:MULTipath:TON {1|0|ON|OFF}
RADar:PULSe:MULTipath:TON?
```

Arguments OFF or 0 disables Multipath. OFF or 0 is the default value.

ON or 1 enables Multipath.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:MULTIPATH:TON 1` enables Multipath for the selected pulse definition.

`RADAR:PULSE:MULTIPATH:TON?` might return `0`, indicating that Multipath is disabled for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:FILE

This command sets or returns the amplitude offset User Defined Filepath for the selected pulse definition.

Conditions Amplitude Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:FILE <filepath>
RADar:PULSe:OFFSets:AMPLitude:FILE?
```

Related Commands RADar:PULSe:OFFSets:AMPLitude:TYPE

Arguments <filepath>::= <string>

Returns A single <filepath> string.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:FILE "C:\AmplitudeOffsets.txt"` sets the Amplitude Offset filepath and filename for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:FILE?` might return `"C:\AmplitudeOffsets.txt"` for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:IOFFset

This command sets or returns the Initial Amplitude Offset value (in dB) when the amplitude offset type is set to Step Offset for the selected pulse definition.

Conditions The amplitude offset type must be set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:IOFFset <initial_offset>
RADar:PULSe:OFFSets:AMPLitude:IOFFset?
```

Related Commands RADar:PULSe:OFFSets:AMPLitude:STEP

Arguments <initial_offset>::= <NRf> value.
Range: –100 dB to 0 dB.

Returns A single <NR2> value.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:IOFFSET –10` sets the Initial Amplitude Offset to –10 dB for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:IOFFSET?` might return `-50.0000000000`, indicating that the Initial Amplitude Offset is set to –50 dB for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:OFFSet

This command sets or returns the Amplitude Offset value (in dB) for the Fixed amplitude offset type for the selected pulse definition.

Conditions The amplitude offset type must be set to Fixed.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:OFFSet <amp_offset>
RADar:PULSe:OFFSets:AMPLitude:OFFSet?
```

Related Commands RADar:PULSe:OFFSets:AMPLitude:TYPE

Arguments <amp_offset>::= <NRf> value.
Range: –100 dB to 0 dB.

Returns A single <NR2> value.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:OFFSET –10` sets the Fixed Amplitude Offset to –10 dB for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:OFFSET ?` might return `-50.0000000000`, indicating that the Fixed Amplitude Offset is set to –50 dB for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:REPeat

This command sets or returns the amplitude offset Repeat Offsets state (enabled or disabled) for the selected pulse definition.

When enabled, if the Pulse Group has a greater repeat count than the offsets listed in the file, the offsets in the file repeat for the rest of the pulses.

Conditions Amplitude Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:REPeat {1|0|ON|OFF}
RADar:PULSe:OFFSets:AMPLitude:REPeat?
```

Arguments OFF or 0 disables the Repeat Offsets state. OFF or 0 is the default value.
ON or 1 enables the Repeat Offsets state.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:REPEAT 1` repeats the offsets listed in the file (if necessary) for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:REPEATRADAR:PULSE:OFFSETS:AMPLITUDE:REPEAT?` might return `0`, indicating that the offsets listed in the file will not be repeated for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:STEP

This command sets or returns the amplitude offset Step Increment value (in dB) for the selected pulse definition.

Conditions Amplitude Offset Type must be set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:STEP <step_increment>
RADar:PULSe:OFFSets:AMPLitude:STEP?
```

Related Commands RADar:PULSe:OFFSets:AMPLitude:TYPE

Arguments <step_increment>::= <NRf> value.
Range: –100 dB to 0 dB.

Returns A single <NR2> value.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:STEP -5` sets the Amplitude Step Increment to –5 dB for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:STEP?` might return `-15.0000000000`, indicating that the Amplitude Step Increment is set to –15 dB for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:TON

This command sets or returns the Amplitude Offset state (enabled or disabled) for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:TON {1|0|ON|OFF}
RADar:PULSe:OFFSets:AMPLitude:TON?
```

Arguments OFF or 0 disables Amplitude Offset. OFF or 0 is the default value.

ON or 1 enables Amplitude Offset.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:AMPLITUDE:TON 1` enables adding Amplitude Offset for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:TON?` might return `0`, indicating that Amplitude Offset is disabled for the selected pulse definition.

RADar:PULSe:OFFSets:AMPLitude:TYPE

This command sets or returns the amplitude offset Type for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:AMPLitude:TYPE {FIXed|SOFFset|UDEFined}
RADar:PULSe:OFFSets:AMPLitude:TYPE?
```

Related Commands RADar:PULSe:OFFSets:AMPLitude:FILE

Arguments

FIXed
SOFFset (Step Offset)
UDEFined (User Defined Offset). This setting requires a file path

Returns

FIX: Fixed
SOFF: Step Offset
UDEF: User Defined Offset

Examples

`RADAR:PULSE:OFFSETS:AMPLITUDE:TYPE FIXED` sets the amplitude offset type to Fixed for the selected pulse definition.

`RADAR:PULSE:OFFSETS:AMPLITUDE:TYPE?` might return `SOFF`, indicating that the amplitude offset type is set to Step Offset for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:FILE

This command sets or returns the frequency offset User Defined Filepath for the selected pulse definition.

Conditions Frequency Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:FILE <filepath>
RADar:PULSe:OFFSets:FREQuency:FILE?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:TYPE

Arguments <filepath>::= <string>

Returns A single <filepath> string.

Examples

`RADAR:PULSE:OFFSETS:FREQUENCY:FILE "C:\Frequency_Offsets.txt"` sets the Frequency Offset filepath and filename for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:FILE?` might return `"C:\Frequency_Offsets.txt"` for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:IOFFset

This command sets or returns the Initial Frequency Offset value (in Hz) when the frequency offset type is set to Step Offset for the selected pulse definition.

Conditions The Frequency Offset Type must me set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:IOFFset <initial_offset>
RADar:PULSe:OFFSets:FREQuency:IOFFset?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:STEP

Arguments <initial_offset>::= <NRf> value.
Range is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:OFFSETS:FREQUENCY:IOFFSET 1E9` sets the Initial Frequency Offset to 1 GHz for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:IOFFSET?` might return `10.0000000000E+6`, indicating that the Initial Frequency Offset is set to 10 MHz for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:OFFSet

This command sets or returns the Frequency Offset value (in Hz) for the Fixed frequency offset type for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:OFFSet <frequency>
RADar:PULSe:OFFSets:FREQuency:OFFSet?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:TYPE

Arguments <frequency>::= <NRf> value.
Range is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:OFFSETS:FREQUENCY:OFFSET 100E3` sets the Fixed Frequency Offset to 100 kHz for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:OFFSET?` might return `1.0000000000E+6`, indicating that the Fixed Frequency Offset is set to 1 MHz for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:REPeat

This command sets or returns the frequency offset Repeat Offsets state (enabled or disabled) for the selected pulse definition.

When enabled, if the Pulse Group has a greater repeat count than the offsets listed in the file, the offsets in the file repeat for the rest of the pulses.

Conditions Frequency Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:REPeat {1|0|ON|OFF}
RADar:PULSe:OFFSets:FREQuency:REPeat?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:TYPE

Arguments OFF or 0 disables the Repeat Offsets state. OFF or 0 is the default value.
ON or 1 enables the Repeat Offsets state.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:FREQUENCY:REPEAT 1` repeats the offsets listed in the file (if necessary) for the selected pulse definition

`RADAR:PULSE:OFFSETS:FREQUENCY:REPEATRADAR:PULSE:OFFSETS:FREQUENCY:REPEAT?` might return `0`, indicating that the offsets listed in the file will not be repeated for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:STEP

This command sets or returns the frequency offset Step Increment value (in Hz) for the selected pulse definition.

Conditions Frequency Offset Type must be set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:STEP <step_increment>
RADar:PULSe:OFFSets:FREQuency:STEP?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:TYPE

Arguments <step_increment>::= <NRf> value.
Range is instrument dependent.

Returns A single <NR3> value.

Examples `RADAR:PULSE:OFFSETS:FREQUENCY:STEP 100E3` sets the Frequency Step Increment to 100 kHz for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:STEP?` might return `1.0000000000E+6`, indicating that the Frequency Step Increment is set to 1 MHz for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:TON

This command sets or returns the Frequency Offset state (enabled or disabled) for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:TON {1|0|ON|OFF}
RADar:PULSe:OFFSets:FREQuency:TON?
```

Arguments OFF or 0 disables Frequency Offset. OFF or 0 is the default value.

ON or 1 enables Frequency Offset.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:FREQUENCY:TON 1` enables adding Frequency Offset for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:TON?` might return `0`, indicating that the Frequency Offset is disabled for the selected pulse definition.

RADar:PULSe:OFFSets:FREQuency:TYPE

This command sets or returns the frequency offset Type for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:FREQuency:TYPE {FIXed|SOFFset|UDEFined}
RADar:PULSe:OFFSets:FREQuency:TYPE?
```

Related Commands RADar:PULSe:OFFSets:FREQuency:FILE

Arguments

FIXed
SOFFset (Step Offset)
UDEFined (User Defined Offset). This setting requires a file path

Returns

FIX: Fixed
SOFF: Step Offset
UDEF: User Defined Offset

Examples

`RADAR:PULSE:OFFSETS:FREQUENCY:TYPE FIXED` changes the frequency offset Type to Fixed for the selected pulse definition.

`RADAR:PULSE:OFFSETS:FREQUENCY:TYPE?` might return `SOFF`, indicating that the frequency offset Type is set to Step Offset for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:FILE

This command sets or returns the phase offset User Defined Filepath for the selected pulse definition.

Conditions Phase Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:FILE <filepath>
RADar:PULSe:OFFSets:PHASe:FILE?
```

Related Commands RADar:PULSe:OFFSets:PHASe:TYPE

Arguments <filepath>::= <string>.

Returns A single <filepath> string.

Examples `RADAR:PULSE:OFFSETS:PHASE:FILE "C:\Phase_Offsets.txt"` sets the Phase Offset filepath and filename for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:FILE?` might return `"C:\Phase_Offsets.txt"` for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:IOFFset

This command sets or returns the Initial Phase Offset value (in degrees) when the phase offset type is set to Step Offset for the selected pulse definition.

Conditions The phase offset Type must me set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:IOFFset <phase_initial_offset>
RADar:PULSe:OFFSets:PHASe:IOFFset?
```

Related Commands RADar:PULSe:OFFSets:PHASe:STEP

Arguments <phase_initial_offset>::= <NRf> value.
Range: –180° to 180°.

Returns A single <NR3> value.

Examples `RADAR:PULSE:OFFSETS:PHASE:IOFFSET 2` sets the Initial Phase Offset of the currently selected path to 2°.

`RADAR:PULSE:OFFSETS:PHASE:IOFFSET?` might return `-5.0000000000`, indicating that the Initial Phase Offset of the currently selected path is set to –5°.

RADar:PULSe:OFFSets:PHASe:OFFSet

This command sets or returns the Phase Offset value (in degrees) for the Fixed phase offset type for the selected pulse definition.

Conditions The phase offset type must be set to Fixed.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:OFFSet <phase_offset>
RADar:PULSe:OFFSets:PHASe:OFFSet?
```

Related Commands RADar:PULSe:OFFSets:PHASe:TYPE

Arguments <phase_offset>::= <NRf> value.
Range: –180° to 180°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:OFFSETS:PHASE:OFFSET 10` sets the Fixed Phase Offset to 10° for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:OFFSET?` might return `-20.0000000000`, indicating that the Fixed Phase Offset is set to –20° for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:REPeat

This command sets or returns the phase offset Repeat Offsets state (enabled or disabled) for the selected pulse definition.

When enabled, if the Pulse Group has a greater repeat count than the offsets listed in the file, the offsets in the file repeat for the rest of the pulses.

Conditions Phase Offset Type must be set to User Defined.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:REPeat {1|0|ON|OFF}
RADar:PULSe:OFFSets:PHASe:REPeat?
```

Arguments OFF or 0 disables the Repeat Offsets state. OFF or 0 is the default value.
ON or 1 enables the Repeat Offsets state.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:PHASE:REPEAT 1` repeats the offsets listed in the file (if necessary) for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:REPEAT?` might return `0`, indicating that the offsets listed in the file will not be repeated for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:STEP

This command sets or returns the phase offset Step Increment value (in degrees) for the selected pulse definition.

Conditions Phase Offset Type must be set to Step Offset.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:STEP <step_increment>
RADar:PULSe:OFFSets:PHASe:STEP?
```

Related Commands RADar:PULSe:OFFSets:PHASe:TYPE

Arguments <step_increment>::= <NRf> value.
Range: –180° to 180°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:OFFSETS:PHASE:STEP -5` sets the Phase Step Increment to –5° for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:STEP?` might return `-5.0000000000`, indicating that the Phase Step Increment is set to –5° for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:TON

This command sets or returns the Phase Offset state (enabled or disabled) for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:TON {1|0|ON|OFF}
RADar:PULSe:OFFSets:PHASe:TON?
```

Arguments OFF or 0 disables Phase Offset. OFF or 0 is the default value.

ON or 1 enables Phase Offset.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:OFFSETS:PHASE:TON 1` enables adding Phase Offset for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:TON?` might return `0`, indicating that Phase Offset is disabled for the selected pulse definition.

RADar:PULSe:OFFSets:PHASe:TYPE

This command sets or returns the phase offset Type for the selected pulse definition.

Group Offsets

Syntax

```
RADar:PULSe:OFFSets:PHASe:TYPE {FIXed|SOFFset|UDEFined}
RADar:PULSe:OFFSets:PHASe:TYPE?
```

Related Commands RADar:PULSe:OFFSets:PHASe:FILE

Arguments

FIXed
SOFFset (Step Offset)
UDEFined (User Defined Offset). This setting requires a file path

Returns

FIX: Fixed
SOFF: Step Offset
UDEF: User Defined Offset

Examples

`RADAR:PULSE:OFFSETS:PHASE:TYPE FIXED` sets the phase offset type to Fixed for the selected pulse definition.

`RADAR:PULSE:OFFSETS:PHASE:TYPE?` might return `SOFF`, indicating that the phase offset type is set to Step Offset for the selected pulse definition.

RADar:PULSe:PENVelope[:CUSTom]:PATH

This command sets or returns the filepath to the custom pulse shape file for the pulse envelope for the selected pulse definition.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope[:CUSTom]:PATH <filepath>
RADar:PULSe:PENVelope[:CUSTom]:PATH?
```

Arguments <filepath>:=<string> defines the path to the custom pulse shape file.

Returns A single <filepath> string.

Examples `RADAR:PULSE:PENVELOPE:CUSTOM:PATH "C:\Customshape.txt"` sets the custom pulse filename and filepath.

`RADAR:PULSE:PENVELOPE:CUSTOM:PATH?` might return `"C:\Customshape.txt"`.

RADar:PULSe:PENVelope[:CUSTom]:SRATe

This command sets or returns the pulse envelope Sampling Rate for the samples defined in the custom pulse shape file for the selected pulse definition.

Conditions Pulse shape must be set to Custom.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope[:CUSTom]:SRATe <sample_rate>
RADar:PULSe:PENVelope[:CUSTom]:SRATe?
```

Arguments <sample_rate>::= <NRf> value.
Range: 1 Hz to 10 GHz.
Default: 10 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:CUSTOM:SRATE 1E9` sets the pulse envelope Sampling Rate to 1 GHz for the samples defined in the custom pulse shape file for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:CUSTOM:SRATE?` might return `5.0000000000E+9`, indicating that the pulse envelope Sampling Rate is set to 1 GHz.

RADar:PULSe:PENVelope:DROOp

This command sets or returns the pulse envelope deviation Droop percentage for the selected pulse definition.

Conditions Droop is only available for pulse shapes Rectangular, Trapezoidal, Raised Cosine, Exponential, and Gaussian.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:DROOp <droop>
RADar:PULSe:PENVelope:DROOp?
```

Arguments <droop>::= <NRf> value.
Range: –50% to 50%.
Default: 0%.

Returns A single <NR2> value.

Examples `RADAR:PULSE:PENVELOPE:DROOP 20` sets the pulse envelope Droop deviation to 20% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:DROOP?` might return `-5.0000000000`, indicating that the pulse envelope Droop deviation is set to –5% for the selected pulse definition.

RADar:PULSe:PENVelope:FTIMe:PERCentage

This command sets or returns the Fall Time Percentage of the pulse envelope for the selected pulse definition.

Conditions Rise Time is only available for pulse shapes Trapezoidal, Raised Cosine, Exponential, and Sawtooth.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:FTIMe:PERCentage
{TNINety|TEIGhty|ZHUNdred}
```

Arguments TNINety: 10% – 90%
TNINety: 10% – 90%
ZHUNdred: 0% – 100%

Returns TNIN: 10% – 90%
TEIG: 10% – 90%
ZHUN: 0% – 100%

Examples `RADAR:PULSE:PENVELOPE:FTIME:PERCENTAGE TNINETY` sets the pulse envelope Fall Time Percentage to 10 to 90% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:FTIME:PERCENTAGE?` might return `ZHUN`, indicating that the pulse envelope Fall Time Percentage is set to 0 to 100% for the selected pulse definition.

RADar:PULSe:PENVelope:FTIMe:VALue

This command sets or returns the Fall Time of the pulse envelope for the selected pulse definition.

Conditions Rise Time is only available for pulse shapes Trapezoidal, Raised Cosine, Exponential, and Sawtooth.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:FTIMe:VALue <fall_time>
RADar:PULSe:PENVelope:FTIMe:VALue
```

Arguments <fall_time>::= <NRf> value.
Range: 2 ns to 100 ms.
Default: 100 ns.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:FTIME:VALUE 1E-6` sets the pulse envelope Fall Time to 1 μs for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:FTIME:VALUE?` might return `100.0000000000E-9`, indicating the pulse envelope Fall Time is set to 100 ns.

RADar:PULSe:PENVelope:OSHoot

This command sets or returns the pulse envelope Overshoot percentage for the selected pulse definition.

Conditions Overshoot is only available for pulse shapes Rectangular, Trapezoidal, Raised Cosine, Exponential, and Gaussian.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:OSHoot <overshoot>
RADar:PULSe:PENVelope:OSHoot?
```

Arguments <overshoot>::= <NRf> value.
Range: 0% to 50%.
Default: 0%.

Returns A single <NR2> value.

Examples `RADAR:PULSE:PENVELOPE:OSHOOT 20` sets the pulse envelope Overshoot to 20% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:OSHOOT?` might return `5.0000000000`, indicating that the pulse envelope Overshoot is set to 5% for the selected pulse definition.

RADar:PULSe:PENVelope:OTIMe

This command sets or returns the pulse envelope Off Time for the selected pulse definition.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:OTIMe <off_time>
RADar:PULSe:PENVelope:OTIMe?
```

Arguments <off_time>::= <NRf> value.
Range: 0 s to 1 s.
Default: 100 µs.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:OTIME 1E-2` sets the pulse envelope Off Time to 10 ms for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:OTIME?` might return `100.0000000000E-6`, indicating that the pulse envelope Off Time is set to 100 µs for the selected pulse definition.

RADar:PULSe:PENVelope:PRF

This command sets or returns the pulse envelope Pulse Repetition Frequency (PRF) for the selected pulse definition.

The PRF value is automatically generated based on the pulse envelope parameters. The PRF value is updated when changing the pulse envelope parameters.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:PRF <repeat_freq>
RADar:PULSe:PENVelope:PRF?
```

Arguments <repeat_freq>::= <NRf> value.
Range and default values are dependent on the pulse envelop parameters and pulse shape.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:PRF 1E3` sets the pulse envelope Pulse Repetition Frequency to 1 kHz for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:PRF?` might return `100.0000000000E+3`, indicating that the pulse envelope Pulse Repetition Frequency is set to 100 kHz.

RADar:PULSe:PENVelope:PRI

This command sets or returns the pulse envelope Pulse Repetition Interval (PRI) for the selected pulse definition.

The PRI value is automatically generated based on the pulse envelope parameters. The PRI value is updated when changing the pulse envelope parameters.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:PRI <repeat_int>
RADar:PULSe:PENVelope:PRI?
```

Arguments <repeat_int>::= <NRf> value.
Range and default values are dependent on the pulse envelop parameters and pulse shape.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:PRI 1E-6` sets the pulse envelope Pulse Repetition Interval to 1 μs for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:PRI?` might return `10.0000000000E-6`, indicating that the pulse envelope Pulse Repetition Interval is set to 10 μs.

RADar:PULSe:PENVelope:RFRequency

This command sets or returns the pulse envelope deviation Ripple Frequency for the selected pulse definition.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:RFRequency <ripple_freq>
RADar:PULSe:PENVelope:RFRequency?
```

Arguments <ripple_freq>::= <NRf> value.
Range: 1 Hz to 25 GHz.
Default: 3 MHz.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:RFREQUENCY 1E2` sets the pulse envelope Ripple Frequency to 100 kHz for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:RFREQUENCY?` might return `1.0000000000E+6`, indicating that the pulse envelope Ripple Frequency is set to 1 MHz for the selected pulse definition.

RADar:PULSe:PENVelope:RIPPle

This command sets or returns the pulse envelope deviation Ripple percentage for the selected pulse definition.

Conditions Ripple is only available for pulse shapes Rectangular, Trapezoidal, Raised Cosine, Exponential, and Gaussian.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:RIPPle <ripple>
RADar:PULSe:PENVelope:RIPPle?
```

Arguments <ripple>::= <NRf> value.
Range: 0% to 50%.
Default: 0%.

Returns A single <NR2> value.

Examples `RADAR:PULSE:PENVELOPE:RIPPLE 20` sets the pulse envelope Ripple to 20% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:RIPPLE?` might return `5.0000000000`, indicating that the pulse envelope Ripple is set to 5% for the selected pulse definition.

RADar:PULSe:PENVelope:RTIMe:PERCentage

This command sets or returns the Rise Time Percentage of the pulse envelope for the selected pulse definition.

Conditions Rise Time is only available for pulse shapes Trapezoidal, Raised Cosine, Exponential, and Sawtooth.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:RTIMe:PERCentage
{TNINety|TEIGhty|ZHUNdred}
```

Arguments

TNINety: 10% – 90%
TNINety: 10% – 90%
ZHUNdred: 0% – 100%

Returns

TNIN: 10% – 90%
TEIG: 10% – 90%
ZHUN: 0% – 100%

Examples

`RADAR:PULSE:PENVELOPE:RTIME:PERCENTAGE TNINETY` sets the pulse envelope Rise Time Percentage to 10 to 90% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:RTIME:PERCENTAGE?` might return `ZHUN`, indicating that the pulse envelope Rise Time Percentage is set to 0 to 100% for the selected pulse definition.

RADar:PULSe:PENVelope:RTIMe:VALue

This command sets or returns the Rise Time of the pulse envelope for the selected pulse definition.

Conditions Rise Time is only available for pulse shapes Trapezoidal, Raised Cosine, Exponential, and Sawtooth.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:RTIMe:VALue <rise_time>
RADar:PULSe:PENVelope:RTIMe:VALue
```

Arguments <rise_time>::= <NRf> value.
Range: 2 ns to 100 ms.
Default: 100 ns.

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:RTIME:VALUE 1E-6` sets the pulse envelope Rise Time to 1 µs for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:RTIME:VALUE?` might return `100.0000000000E-9`, indicating the pulse envelope Rise Time is set to 100 ns.

RADar:PULSe:PENVelope:SHAPe

This command sets or returns the Shape of the pulse envelope for the selected pulse definition.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:SHAPe
{RECTangular|TRAPezoidal|RCOSine|EXPonential|STOOth|GAUSian
|CUSTom}
```

Arguments RECTangular, TRAPezoidal, RCOSine, EXPonential, STOOth, GAUSian, and CUSTom are the selectable pulse envelope types.
Default: Trapezoidal.

Returns
RECT: Rectangular
TRAP: Trapezoidal
RCOS: Raised cosine
EXP: Exponential
STOO: Sawtooth
GAUS: Gausian
CUST: Custom

Examples `RADAR:PULSE:PENVELOPE:SHAPE RECTANGULAR` sets the pulse shape to Rectangular for the pulse definition.

`RADAR:PULSE:PENVELOPE:SHAPE?` might return `GAUS`, indicating that the pulse shape is set to Gausian for the pulse definition.

RADar:PULSe:PENVelope:WIDTh:PERCentage

This command sets or returns the percentage at which the pulse width value is applied to the pulse shape for the pulse envelope for the selected pulse definition.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:WIDTh:PERCentage {FIFTy|HUNDred}
RADar:PULSe:PENVelope:WIDTh:PERCentage?
```

Arguments

FIFTy: 50 percent.
HUNDred: 100 percent.

Returns

FIFT: 50 percent.
HUND: 100 percent.

Examples

`RADAR:PULSE:PENVELOPE:WIDTH:PERCENTAGE HUNDRED` sets the pulse envelope Pulse Width percentage to 100% for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:WIDTH:PERCENTAGE?` might return `FIFT`, indicating that the pulse envelope Pulse Width percentage is set to 50% for the selected pulse definition.

RADar:PULSe:PENVelope:WIDTh:VALue

This command sets or returns the Pulse Width of the pulse envelope for the selected pulse definition.

Conditions Pulse Width is only available for pulse shapes Rectangular, Trapezoidal, Raised Cosine, Exponential, and Gaussian.

Group Pulse envelope

Syntax

```
RADar:PULSe:PENVelope:WIDTh:VALue <pulse_width>
RADar:PULSe:PENVelope:WIDTh:VALue?
```

Arguments <pulse_width>::= <NRf> value.
Range: 200 ps to 1 s. The minimum pulse width is 100 ns when the Pulse Width percentage is set to 50%.
Default: 1 µs

Returns A single <NR3> value.

Examples `RADAR:PULSE:PENVELOPE:WIDTH:VALUE 1E-2` sets the pulse envelope Pulse Width to 10 ms for the selected pulse definition.

`RADAR:PULSE:PENVELOPE:WIDTH:VALUE?` might return `100.0000000000E-9`, indicating that the pulse envelope Pulse Width is set to 100 ns for the selected pulse definition.

RADar:PULSe:REPeat

This command sets or returns the Repeat count of the currently selected pulse definition.

Group Pulse train

Syntax

```
RADar:PULSe:REPeat <repeat_count>
RADar:PULSe:REPeat?
```

Arguments <repeat_count>::= <NR1> value.
Default: 1.

Returns A single <NR1> value.

Examples `RADAR:PULSE:REPEAT 10` sets the Repeat Count of the pulse definition to 10.

`RADAR:PULSE:REPEAT?` might return `5`, indicating that the Repeat Count of the pulse definition is set to 5.

RADar:PULSe:SELect

This command sets or returns the index of the currently selected pulse definition in the currently selected pulse train. Other commands act on the selected pulse definition.

Group Pulse train

Syntax

```
RADar:PULSe:SELect <index>
RADar:PULSe:SELect?
```

Arguments <index>::=<NR1> value is a valid pulse definition index.

Returns A single <NR1> value.

Examples `RADAR:PULSE:SELECT 2` selects pulse definition index 2 of the currently selected pulse train.

`RADAR:PULSE:SELECT?` might return `4`, indicating that index four is the currently selected pulse definition of the selected pulse train.

RADar:PULSe:SPRI:DTYPe

This command sets or returns the Staggered Pulse Repetition Interval deviation type for the selected pulse definition.

Conditions When setting the type to File,

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:DTYPe {RAMP|FILE|TABLe}
RADar:PULSe:SPRI:DTYPe?
```

Arguments

RAMP
FILE (Requires file path command)
TABLe

Returns

RAMP
FILE
TABL

Examples

`RADAR:PULSE:SPRI:DTYPE RAMP` sets the Staggered PRI type to Ramp for the selected pulse definition.

`RADAR:PULSE:SPRI:DTYPE?` might return `FILE`, indicating that the Staggered PRI is set to use a deviation file for the selected pulse definition.

RADar:PULSe:SPRI:FILE

This command sets or returns the filepath to the Staggered PRI Deviation file for the selected pulse definition.

Conditions The Deviation Type must be set to File.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:FILE <filepath>
RADar:PULSe:SPRI:FILE?
```

Related Commands RADar:PULSe:SPRI:DTYPe

Arguments <filepath>::=<string> defines the path to the deviation file.

Returns A single <filepath> string.

Examples `RADAR:PULSE:SPRI:FILE "C:\DeviationFile.txt"` sets the deviation filepath and filename for the selected pulse definition.

`RADAR:PULSE:SPRI:FILE?` might return `"C:\DeviationFile.txt"`.

RADar:PULSe:SPRI:RAMP:DEGRees

This command sets or returns the Ramp Degrees for the PRI Deviation slope for the selected pulse definition.

Setting the ramp Degrees value enables Ramp Deviation and disables Minimum Deviation for the Ramp Deviation.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:RAMP:DEGRees <degrees>
RADar:PULSe:SPRI:RAMP:DEGRees?
```

Related Commands RADar:PULSe:SPRI:RAMP:MDEViation

Arguments <degrees>::= <NR3> value.

Range: 0° to 45°.

Returns A single <NR2> value.

Examples `RADAR:PULSE:SPRI:RAMP:DEGREES 3` sets the Ramp slope degrees to 3 degrees for the selected pulse definition.

`RADAR:PULSE:SPRI:RAMP:DEGREES?` might return `2.0000000000`, indicating that the Ramp slope degrees is set to 2 degrees for the selected pulse definition.

RADar:PULSe:SPRI:RAMP:MDEViation

This command sets or returns the Ramp Minimum Deviation (ΔT) for the PRI Deviation slope for the selected pulse definition.

Setting the minimum deviation value enables Minimum Deviation and disables Degrees for the Ramp Deviation.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:RAMP:MDEViation <min_deviation>
RADar:PULSe:SPRI:RAMP:MDEViation?
```

Related Commands RADar:PULSe:SPRI:RAMP:DEGRees

Arguments <min_deviation>::= <NRf> value.

Range: 0 s to PRI.
Default: 0 s.

Returns A single <NR1> value.

Examples `RADAR:PULSE:SPRI:RAMP:MDEVIATION 1E-3` sets the Ramp minimum deviation to 1 ms for the selected pulse definition.

`RADAR:PULSE:SPRI:RAMP:MDEVIATION?` might return `10.0000000000E-6`, indicating that the minimum Ramp deviation is set to 10 µs for the selected pulse definition.

RADar:PULSe:SPRI:RAMP:SLOPe

This command sets or returns the ramp Slope direction for the Staggered PRI Deviation ramp for the selected pulse definition.

Group Pulse repetition

Syntax `RADar:PULSe:SPRI:RAMP:SLOPe {UP|DOWN|UDOWn|DUP}`

Arguments RADar:PULSe:SPRI:RAMP:MDEViation, RADar:PULSe:SPRI:RAMP:DEGRees

UP
DOWN
UDOWn: Up Down
DUP: Down Up

Returns UP
DOWN
UDOW: Up Down
DUP: Down Up

Examples `RADAR:PULSE:SPRI:RAMP:SLOPE DOWN` sets the ramp angle direction to down for the selected pulse definition.

`RADAR:PULSE:SPRI:RAMP:SLOPE?` might return `UP` indicating that the ramp slope direction is angled up for the selected pulse definition.

RADar:PULSe:SPRI:REPeat

This command sets or returns the Repeat deviation pattern (enabled or disabled) for the Staggered PRI for the selected pulse definition. When enabled, the PRI Deviation pattern repeats for the indices that have been left blank.

Conditions This is only available when the Deviation Type is set to FILE or TABLE.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:REPeat {1|0|ON|OFF}
RADar:PULSe:SPRI:REPeat?
```

Arguments OFF or 0 disables Repeat deviation pattern. OFF or 0 is the default value.

ON or 1 enables Repeat deviation pattern.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:SPRI:REPEAT 1` enables repeating PRI Deviation pattern for the selected pulse definition.

`RADAR:PULSE:SPRI:REPEAT?` might return `0`, indicating that the PRI Deviation pattern will not be repeated for the selected pulse definition.

RADar:PULSe:SPRI:TABLe:ADD (No Query Form)

This command adds the specified number of repeat indices to the end of the user defined table of Staggered PRI for the selected pulse group.

The total number of repeat indices cannot exceed 100. A warning occurs if the number of indices exceeds the set Repeat count for the selected pulse definition.

Conditions The staggered PRI Deviation Type must be set to Table.

Group Pulse repetition

Syntax `RADar:PULSe:SPRI:TABLe:ADD <count>`

Related Commands RADar:PULSe:SPRI:TABLe:COUNt?

Arguments <count>::= <NR1> value.

Examples `RADAR:PULSE:SPRI:TABLE:ADD 3` adds 3 indices to the user defined table for Staggered PRI for the selected pulse group.

RADar:PULSe:SPRI:TABLe:COUNt? (Query Only)

This command returns the number of repeat indices in the user defined table of Staggered PRI for the selected pulse group.

Group Pulse repetition

Syntax `RADar:PULSe:SPRI:TABLe:COUNt?`

Related Commands RADar:PULSe:REPeat

Returns A single <NR1> value.

Examples `RADAR:PULSE:SPRI:TABLE:COUNT?` might return `5`, indicating there are 5 indices defined in the user defined Staggared PRI Diviation table for the selected pulse definition.

RADar:PULSe:SPRI:TABLe:INDex[n]:DELete (No Query Form)

This command deletes the specified number of repeat indices, starting with the specified index location in the user defined table of Staggered PRI for the selected pulse group.

Group Pulse repetition

Syntax `RADar:PULSe:SPRI:TABLe:INDex[n]:DELete <count>`

Arguments [n]::= <NR1> (“n” determines the index number in the user defined deviation table of staggered PRI).

If omitted, n is interpreted as 1.

<count>::= <NR1> value.

Examples `RADAR:PULSE:SPRI:TABLE:INDEX10:DELETE 5` deletes the 5 indices in the deviation table, starting with step index number 10. In this example, steps 10 through 14 are deleted.

RADar:PULSe:SPRI:TABLe:INDex[n]:DEViation

This command sets or returns the Deviation of the specified index location in the user defined table of Staggered PRI for the selected pulse group.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:TABLe:INDex[n]:DEViation <deviation>
RADar:PULSe:SPRI:TABLe:INDex[n]:DEViation?
```

Arguments [n]::= <NR1> (“n” determines the step index number in the user defined deviation table of staggered PRI).

If omitted, n is interpreted as 1.

<deviation>::= <NRf> value.
Range: –100 ms to 100 ms.
Defualt: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:SPRI:TABLE:INDEX3:DEVIATION 1E-12` sets the deviation at index 3 of the user defined table to 1 ps of Staggered PRI for the selected pulse group.

`RADAR:PULSE:SPRI:TABLE:INDEX3:DEVIATION?` might return `65.0000000000E-12`, indicating that the deviation at index 3 of the user defined table is set to 65 ps for Staggered PRI for the selected pulse group.

RADar:PULSe:SPRI:TABLe:INDex[n]:INSert (No Query Form)

This command inserts the specified number of repeat indices above the specified index location in the user defined table of Staggered PRI for the selected pulse group.

Conditions If no indices exist in the deviation table, use the Add command to add indices.

Group Pulse repetition

Syntax `RADar:PULSe:SPRI:TABLe:INDex[n]:INSert <count>`

Related Commands RADar:PULSe:SPRI:TABLe:ADD

Arguments [n]::= <NR1> (“n” determines the index number in the user defined deviation table of staggered PRI).

If omitted, n is interpreted as 1.

<count>::= <NR1> value.

Examples `RADAR:PULSE:SPRI:TABLE:INDEX3:INSERT 2` inserts two indices in front of the existing 3rd index in the user defined table of Staggered PRI for the selected pulse group.

RADar:PULSe:SPRI:TON

This command sets or returns the Staggered Pulse Repetition Interval state (enabled or disabled) for the selected pulse group.

Group Pulse repetition

Syntax

```
RADar:PULSe:SPRI:TON {0|1|OFF|ON}
RADar:PULSe:SPRI:TON?
```

Arguments OFF or 0 disables Staggered PRI. OFF or 0 is the default value.
ON or 1 enables Staggered PRI.

Returns A single <Boolean> value.

Examples `RADAR:PULSE:SPRI:TON ON` enables Staggered PRI for the selected pulse group.

`RADAR:PULSE:SPRI:TON?` might return `1`, indicating that Staggered PRI is enabled for the selected pulse group.

RADar:PULSe:STIMe

This command sets or returns the Start Time of the currently selected pulse definition.

Group Pulse train

Syntax

```
RADar:PULSe:STIMe <start_time>
RADar:PULSe:STIMe?
```

Arguments <start_time>::= <NRf> value.
Range: 0 s to 1 s.
Default: 0 s.

Returns A single <NR3> value.

Examples `RADAR:PULSE:STIME 12E-6` sets the Start Time of the pulse definition to 12 µs.

`RADAR:PULSE:STIME?` might return `1.0000000000E-12`, indicating that the Start Time of the pulse definition is set to 1 ps.

RADar:PULSe:TYPe

This command sets or returns the type of the pulse definition for the selected pulse definition.

Group Pulse train

Syntax

```
RADar:PULSe:TYPe {PULSed|DTIMe}
RADar:PULSe:TYPe?
```

Arguments {PULSed: Pulse group definition.

DTIMe: Dead time.

Returns PULS: Pulse group.
DTIM: Dead time.

"" is returned if the type is Combined Pulse.

Examples `RADAR:PULSE:TYPE DTIMe` sets the pulse definition as a dead time for the currently selected definition.

`RADAR:PULSE:TYPE?` might return `PULS`, indicating that the pulse definition is set to pulsed for the currently selected pulse definition.

RADar:RESet (No Query Form)

This command resets the Radar Signals plug-in by setting all the values within the module to their default values.

Group Control

Syntax `RADar:RESet`

Examples `RADAR:RESET` returns the Radar Signals plug-in to its default values.

RADar:SFORmat

This command sets or returns the signal format of the signal to be compiled for all pulse trains.

Group Compile

Syntax

```
RADar:SFORmat {RF|IQ}
RADar:SFORmat?
```

Arguments RF sets the signal format to RF/IF.

IQ sets the signal format to IQ.

Returns

RF
IQ

Examples `RADAR:SFORMAT IQ` sets the signal format to IQ mode for all pulse trains.

`RADAR:SFORMAT?` might return `RF`, indicating that the signal format is set to RF/IF mode for all pulse trains.

RADar:SPARameter:CASCading:STAGe[m][:ENABle]

This command sets or returns the state of the specified Cascaded S-Parameter stage (enabled or disabled) for the selected pulse train.

Conditions S-Parameter Mode must be set to Cascading.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m][:ENABle] {0|1|OFF|ON}
RADar:SPARameter:CASCading:STAGe[m][:ENABle]?
```

Related Commands RADar:SPARameter:MODE
RADar:SPARameter:NCAScading:FILE

Arguments [m] ::= {1 – 6} ("m" determines the stage number)

If omitted, m is interpreted as 1.

OFF or 0 disables the specified Cascading Stage. OFF or 0 is the default value. ON or 1 enables the Stage.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:CASCADING:STAGE6:ENABLE 1` enables Stage 6 in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE6:ENABLE?` might return `0`, indicating that Stage 6 is not enabled in Cascading mode for the selected pulse train.

RADar:SPARameter:MODE

This command sets or returns the S-Parameter mode (Cascading or Non-Cascading) for the selected pulse train.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:MODE {CASC|NCAS}
RADar:SPARameter:MODE?
```

Arguments

`CASCading` sets the S-Parameter mode to cascading. allowing you to cascade up to six S-parameter files and apply the characteristics on the waveform.

`NCASCading` sets the S-Parameter mode to non-cascading, allowing you to apply S-parameter characteristics on the waveform from only one S-parameter file.

Returns

`CASC`

`NCASC`

Examples

`RADAR:SPARAMETER:MODE CASCADING` sets the S-Parameter mode to cascading for the selected pulse train.

`RADAR:SPARAMETER:MODE?` might return `NCAS`, indicating that the S-Parameter mode is set to Non-Cascading mode for the selected pulse train.

RADar:SPARameter:SFORmat

This command sets or returns the signal format (I or Q) for all S-Parameter values for all pulse trains when the Radar plug-in signal format is set to IQ.

Conditions Requires an S-Parameters license.

The signal format for the Radar plug-in must be set to IQ.

Group S-Parameters

Syntax

```
RADar:SPARameter:SFORmat {I|Q}
RADar:SPARameter:SFORmat?
```

Related Commands RADar:SFORmat

Arguments I sets the signal format to IQ and all S-Parameter settings apply to the Q signal.

Q sets the signal format to IQ and all S-Parameter settings apply to the I signal.

Returns RF
I
Q

Examples `RADAR:SPARAMETER:SFORMAT Q` applies the S-Parameters to the Q signal for the selected pulse train.

`RADAR:SPARAMETER:SFORMAT?` might return `I`, indicating that all S-Parameters values apply to the I signal for the selected pulse train.

RADar:SPARameter:SFORmat:LIQ

This command sets or returns the S-Parameter Couple Settings (I,Q) state. When enabled, all I and Q S-Parameters are linked together (chained) so that all parameters match between I and Q for the selected pulse train.

When first enabled, all Q inherit the I settings. Any changes after enabling coupling affects both I and Q settings.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:SFORmat:LIQ {1|0|ON|OFF}
RADar:SPARameter:SFORmat:LIQ?
```

Related Commands RADar:SPARameter:SFORmat

Arguments OFF or 0 disables Couple Settings. OFF or 0 is the default value.

ON or 1 enables Couple Settings.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:SFORMAT:LIQ 1` enables the Couple Settings for the selected pulse train.

`RADAR:SPARAMETER:SFORMAT:LIQ?` might return `0`, indicating that the Couple Settings is disabled for the selected pulse train.

RADar:SPARameter:TON

This command sets or returns the S-Parameter state (enabled or disabled) for the selected pulse train.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:TON {1|0|ON|OFF}
RADar:SPARameter:TON?
```

Arguments OFF or 0 disables S-Parameters. OFF or 0 is the default value.

ON or 1 enables S-Parameters.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:TON 1` enables the S-Parameters for the selected pulse train.

`RADAR:SPARAMETER:TON?` might return `0`, indicating that the S-Parameters is disabled for the selected pulse train.

RADar:SPARameter:BANDwidth

This command sets or returns the S-Parameter bandwidth when setting manually for the selected pulse train.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:BANDwidth {FULL|<bandwidth>}
RADar:SPARameter:BANDwidth?
```

Related Commands RADar:SPARameter:BANDwidth:AUTO

Arguments FULL – The bandwidth is set to ½ of the waveform's sample rate (i.e. Nyquist Frequency).

<bandwidth>::= <NR3> value.

Range: 1 Hz to ½ of the maximum sample rate of the instrument.

If the set Bandwidth is greater than the Nyquist (Sample rate of the waveform/2), then the software limits the bandwidth to ½ of the waveform's sample rate.

Returns FULL
A single <NR3> value.

Examples `RADAR:SPARAMETER:BANDWIDTH 60E6` sets the S-Parameter Bandwidth Value to 60 MHz for the selected pulse train.

`RADAR:SPARAMETER:BANDWIDTH?` might return `1.0000000000E+9`, indicating the S-Parameter Bandwidth is set to 1 GHz for the selected pulse train.

RADar:SPARameter:BANDwidth:AUTO

This command sets or returns the S-Parameter automatic bandwidth calculation setting for the selected pulse train.

The bandwidth is defined at the point where the signal rolls off to –60 dB. If this results in a bandwidth greater than the instrument supports, the bandwidth is set to ½ of the waveform's sample rate (i.e. Nyquist Frequency).

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:BANDwidth:AUTO {0|1|OFF|ON}
RADar:SPARameter:BANDwidth:AUTO?
```

Related Commands RADar:SPARameter:BANDwidth

Arguments ON or 1 enables automatic bandwidth calculation. ON or 1 is the default value.

OFF or 0 disables automatic bandwidth calculation and sets the Bandwidth setting to Manual, requiring a value.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:BANDWIDTH:AUTO 0` disables the S-Parameter automatic bandwidth calculation and sets it to use a manual value for the selected pulse train.

`RADAR:SPARAMETER:BANDWIDTH:AUTO?` might return 1, indicating the S-Parameter automatic bandwidth calculation is enabled for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor2[:ENABle]

This command sets or returns the aggressor 2 signal type state (enabled or disabled) in Cascading mode for the selected pulse train. Aggressor 2 signals are available when the number of ports is set to 12.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration must be set to 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor2[:ENABle] {0|1|ON|OFF}
RADar:SPARameter:CASCading:AGGRessor2[:ENABle]?
```

Related Commands

RADar:SPARameter:MODE

Arguments

OFF or 0 disables the aggressor 2 signal type. OFF or 0 is the default value. ON or 1 enables the aggressor 2 signal type.

Returns

A single <Boolean> value.

Examples

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:ENABLE ON` enables the aggressor 2 signal type, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:ENABLE?` might return `0`, indicating that the aggressor 2 signal type is disabled, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:AMPLitude

This command sets or returns the specified Aggressor's amplitude, in Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:AMPLitude
<amplitude>
RADar:SPARameter:CASCading:AGGRessor[n]:AMPLitude?
```

Related Commands RADar:SPARameter:MODE

Arguments [n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<amplitude> ::= <NRf>

Returns A single <NR3> value.

Examples `RADAR:SPARAMETER:CASCADING:AGGRESSOR1:AMPLITUDE 200E-3` sets the first Aggressor's amplitude to 200 mV, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:AMPLITUDE?` might return `100.0000000000E-3`, indicating that the 2nd Aggressor's amplitude is set to 100 mV, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:CTALk

This command sets or returns the specified Aggressor's crosstalk type, in Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:CTALk
{NEXT|FEXT|BOTH}
RADar:SPARameter:CASCading:AGGRessor[n]:CTALk?
```

Related Commands

RADar:SPARameter:MODE

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

NEXT – Near-End Crosstalk

FEXT – Far-End Crosstalk

BOTH – Near and Far-End Crosstalk

Returns

NEXT
FEXT
BOTH

Examples

`RADAR:SPARAMETER:CASCADING:AGGRESSOR1:CTALK FEXT` sets the first Aggressor's Crosstalk type to Far End Crosstalk, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:CTALK?` might return `NEXT`, indicating that the 2nd Aggressor crosstalk type is set to Far End Crosstalk, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:DRATe

This command sets or returns the specified Aggressor's data rate, in Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:DRATe <data_rate>
RADar:SPARameter:CASCading:AGGRessor[n]:DRATe?
```

Related Commands RADar:SPARameter:MODE

Arguments [n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<data_rate> ::= <NRf>

Returns A single <NR3> value.

Examples `RADAR:SPARAMETER:CASCADING:AGGRESSOR1:DRATE 4E9` sets the first Aggressor's data rate to 4 Gbps, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:DRATE?` might return `500.0000000000E+3`, indicating that the 2nd Aggressor's data rate is set to 500 kbps, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal

This command sets or returns specified Aggressor's signal type, in Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal
{CLOCk|PRBS|FILE|SAVictim}
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

CLOCk – Indicates that the aggressor signal is a clock pattern.

PRBS – Indicates that the aggressor signal is a PBRS pattern. You also must set the PBRS type.

FILE – Aggressor is set to use a file. You must set the file path.

SAVictim – Aggressor is the same as the victim.

Returns

CLOC
PRBS
FILE
SAV

Examples

`RADAR:SPARAMETER:CASCADING:AGGRESSOR1:SIGNAL SAVICTIM` sets the aggressor signal to be the same as the victim, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:SIGNAL?` might return `FILE`, indicating that 2nd Aggressor has a signal type set to use a file, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE

This command sets or returns the filepath to the aggressor file for the specified Aggressor, in Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Aggressor signal type must be File.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE
<filepath>
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<filepath> ::= <string> defines the path to the aggressor file.

Returns

A single <filepath> string.

Examples

`RADAR:SPARAMETER:CASCADING:AGGRESSOR1:SIGNAL:FILE "C:\temp\myFile.txt"` sets the first Aggressor's file and filepath when the aggressor is set to use a file, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:SIGNAL:FILE?` might return `"C:\temp\myFile.txt"` indicating that the 2nd Aggressor has a signal type filepath set to "C:\temp\myFile.txt", in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:PRBS

This command sets or returns the specified Aggressor's PRBS signal type, in Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration muse be set to 8 or 12 ports.

Aggressor signal type must be PRBS.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:PRBS
{PRBS7|PRBS9|PRBS15|PRBS16|PRBS20|PRBS21|PRBS23|PRBS29|PRBS31}
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:PRBS?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

Patterns available include: PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31.

Returns

PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31

Examples

`RADAR:SPARAMETER:CASCADING:AGGRESSOR1:SIGNAL:PRBS PRBS31` sets the first Aggressor's Signal type's PRBS value to PRBS31, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:AGGRESSOR2:SIGNAL:PRBS?` might return `PRBS15`, indicating that the 2nd Aggressor has a signal type PRBS value set to PRBS15, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:DEEMbed

This command sets or returns whether the Cascading S-Parameters is to de-embed (invert) the S-Parameters, in Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Cascading.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:DEEMbed {0|1|OFF|ON}
RADar:SPARameter:CASCading:DEEMbed?
```

Related Commands RADar:SPARameter:MODE

Arguments OFF or 0 disables de-embedding. OFF or 0 is the default value.
ON or 1 enables de-embedding.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:CASCADING:DEEMBED 1` will de-embed the S-Parameters for Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:DEEMBED?` might return `0`, indicating that S-Parameters will not be de-embedded for Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:DRX[n]

This command sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified receiver port number (Rx-Port) in Cascading mode and Differential Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

S-Parameter Signalling Scheme must be set to Differential (where applicable).

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:DRX[n] <port number>
RADar:SPARameter:CASCading:STAGe[m]:DRX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:CASCading:TYPE
RADar:SPARameter:CASCading:STAGe[m]:DTX[n]

Arguments

[m] ::= {1|2|3|4|5|6}. A variable value to define the Stage.

If omitted, interpreted as 1

[n] ::= <NR1> value. A variable value to define the receiver port number (Rx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4, then n = {1}
Type = 8 then n = {1 – 2}
Type = 12 then n = {1 – 3}

If omitted, n is interpreted as 1.

<port number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Rx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4 then <port number> = {1 – 2}
Type = 8 then <port number> = {1 – 4}
Type = 12 then <port number> = {1 – 6}

Returns

A single <NR1> value.

Examples

`RADAR:SPARAMETER:CASCADING:STAGE2:DRX2 4` assigns S-Parameter port 4 to the channel's receiver port 2 for Stage 2, in the Differential, Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE6:RX3?` might return `10`, indicating that S-Parameter Port 10 is assigned to the channel's receiver port 3 for Stage 6, in the Differential, Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:DTX[n]

This command sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified transmission port number (Tx-Port) in Cascading mode and Differential Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

S-Parameter Signalling Scheme must be set to Differential (where applicable).

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:DTX[n] <port number>
RADar:SPARameter:CASCading:STAGe[m]:DTX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:CASCading:TYPE
RADar:SPARameter:CASCading:STAGe[m]:DRX[n]

Arguments

[m] ::= {1|2|3|4|5|6}. A variable value to define the Stage.

If omitted, interpreted as 1

[n] ::= <NR1> value. A variable value to define the transmission port number (Tx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4, then n = {1}
Type = 8 then n = {1 – 2}
Type = 12 then n = {1 – 3}

If omitted, n is interpreted as 1.

<port number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Tx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4 then <port number> = {1 – 2}
Type = 8 then <port number> = {1 – 4}
Type = 12 then <port number> = {1 – 6}

Returns

A single <NR1> value.

Examples

`RADAR:SPARAMETER:CASCADING:STAGE2:DTX2 4` assigns S-Parameter port 4 to the channel's transmission port 2 for Stage 2, in the Differential, Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE6:TX3?` might return `10`, indicating that S-Parameter Port 10 is assigned to the channel's transmission port 3 for Stage 6, in the Differential, Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:FILE

This command sets or returns the filepath for the specified S-Parameters Cascading Stage, in Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Cascading.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:FILE <filepath>
RADar:SPARameter:CASCading:STAGe[m]:FILE
```

Related Commands RADar:SPARameter:MODE

Arguments [m] ::= {1 – 6} ("m" determines the stage number)

If omitted, m is interpreted as 1.

<filepath>::= <string> defines the path to the S-Parameter file.

Returns <filepath>::= <string>.

Examples `RADAR:SPARAMETER:CASCADING:STAGE1:FILE "C:\temp\myFile.s12p"` sets the filepath to "C:\temp\myFile.s12p" for use during compilation for Stage 1 for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE1:FILE?` might return `"C:\temp\myFile.s12p"` indicating the filepath for Stage 1 for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:RX[n]

This command sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified receiver port number (Rx-Port) in Cascading mode and Single-Ended Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

S-Parameter Signalling Scheme must be set to Single-Ended (where applicable).

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:RX[n] <port_number>
RADar:SPARameter:CASCading:STAGe[m]:RX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:CASCading:TYPE
RADar:SPARameter:CASCading:STAGe[m]:TX[n]

Arguments

[m] ::= {1|2|3|4|5|6}. A variable value to define the Stage.

If omitted, interpreted as 1

[n] ::= <NR1> value. A variable value to define the receiver port number (Rx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then n = no value
Type = 2, then n = {1}
Type = 4 then n = {1 – 2}
Type = 6 then n = {1 – 3}
Type = 8 then n = {1 – 4}
Type = 12 then n = {1 – 6}

<port_number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Tx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then <port number> = no value
Type = 2, then <port number> = {1 – 2}
Type = 4 then <port number> = {1 – 4}
Type = 6 then <port number> = {1 – 6}

Type = 8 then <port number> = {1 – 8}
Type = 12 then <port number> = {1 – 12}

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:CASCADING:STAGE2:RX2 4` assigns S-Parameter port 4 to the channel's receiver port 2 for Stage 2, in the Single-Ended, Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE6:RX3?` might return `10`, indicating that S-Parameter Port 10 is assigned to the channel's receiver port 3 for Stage 6, in the Single-Ended, Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:SSCHeme

This command sets or returns the S-Parameter Signalling Scheme (single-ended or differential), in Cascading mode. Signalling Scheme is only available for 4, 8, or 12 port configurations for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration must be 4, 8, or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:SSCHeme
{SENDed|DIFFerential}
RADar:SPARameter:CASCading:STAGe[m]:SSCHeme?
```

Related Commands

RADar:SPARameter:MODE,
RADar:SPARameter:CASCading:TYPE

Arguments

[m] ::= {1 – 6} ("m" determines the stage number)

If omitted, m is interpreted as 1.

SENDed – Single Ended Signal Scheme

DIFFerential – Differential Signal Scheme

Returns

SEND
DIFF

Examples

`RADAR:SPARAMETER:CASCADING:STAGE2:SSCHEME DIFF` sets the Stage 2 Signalling Scheme to Differential, in Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE3:SSCHEME?` might return `SEND`, indicating that the Stage 3 Signalling Scheme is set to Single Ended, in Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STAGe[m]:TX[n]

This command sets or returns the S-Parameter port assignment of the specified Stage and the channel's specified transmission port number (Tx-Port) in Cascading mode and Single-Ended Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Cascading.

S-Parameter Signalling Scheme must be set to Single-Ended (where applicable).

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STAGe[m]:TX[n] <port number>
RADar:SPARameter:CASCading:STAGe[m]:TX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:CASCading:TYPE
RADar:SPARameter:CASCading:STAGe[m]:RX[n]

Arguments

[m] ::= {1|2|3|4|5|6}. A variable value to define the Stage.

If omitted, interpreted as 1

[n] ::= <NR1> value. A variable value to define the transmission port number (Tx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then n = no value
Type = 2, then n = {1}
Type = 4 then n = {1 – 2}
Type = 6 then n = {1 – 3}
Type = 8 then n = {1 – 4}
Type = 12 then n = {1 – 6}

<port number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Tx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then <port number> = no value
Type = 2, then <port number> = {1 – 2}
Type = 4 then <port number> = {1 – 4}
Type = 6 then <port number> = {1 – 6}

Type = 8 then <port number> = {1 – 8}
Type = 12 then <port number> = {1 – 12}

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:CASCADING:STAGE2:TX2 4` assigns S-Parameter port 4 to the channel's transmission port 2 for Stage 2, in the Single-Ended, Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STAGE6:TX3?` might return `10`, indicating that S-Parameter Port 10 is assigned to the channel's transmission port 3 for Stage 6, in the Single-Ended, Cascading mode for the selected pulse train.

RADar:SPARameter:CASCading:STYPe

This command sets or returns S-Parameter signal type (victim or aggressor), in Cascading mode for the selected pulse train for the selected pulse train. The number of ports must be either 8 or 12.

Conditions

S-Parameter Mode must be set to Cascading.

Port configuration must be 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:CASCading:STYPe {VICTim|AGGRessor|BOTH}
RADar:SPARameter:CASCading:STYPe?
```

Related Commands

RADar:SPARameter:MODE

Arguments

VICTim – enables the victim signal type.

AGGRessor – enables the aggressor signal type.

BOTH – enables the victim and aggressor signal types.

Returns

VICT
AGGR
BOTH

Examples

`RADAR:SPARAMETER:CASCADING:STYPE BOTH` sets the signal type to include both the Victim and Aggressor signal types, in Cascading mode for the selected pulse train for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:STYPE?` might return `AGGR`, indicating that the S-Parameter signal type is currently set to be Aggressor, in Cascading mode for the selected pulse train for the selected pulse train.

RADar:SPARameter:CASCading:TYPE

This command sets or returns the S-Parameter number of ports, in Cascading mode for the selected pulse train.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:CASCading:TYPE {2|4|6|8|12}
RADar:SPARameter:CASCading:TYPE?
```

Arguments {2|4|6|8|12} – defines the number of S-Parameter ports.

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:CASCADING:TYPE 12` sets the S-Parameter type to a 12-Port system for the cascading mode for the selected pulse train.

`RADAR:SPARAMETER:CASCADING:TYPE?` might return `6`, indicating that the S-Parameter type is a 6-Port system for Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor2[:ENABle]

This command sets or returns the aggressor 2 signal type state (enabled or disabled) in Non-Cascading mode for the selected pulse train. Aggressor 2 signals are available when the number of ports is set to 12.

Conditions S-Parameter Mode must be set to Non-Cascading.

Port configuration must be set to 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor2[:ENABle] {0|1|ON|OFF}
RADar:SPARameter:NCAScading:AGGRessor2[:ENABle]?
```

Related Commands RADar:SPARameter:MODE

Arguments OFF or 0 disables the aggressor 2 signal type. OFF or 0 is the default value. ON or 1 enables the aggressor 2 signal type.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:ENABLE ON` enables the aggressor 2 signal type, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:ENABLE?` might return `0`, indicating that the aggressor 2 signal type is disabled, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:AMPLitude

This command sets or returns the specified Aggressor's amplitude, in Non-Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:AMPLitude
<amplitude>
RADar:SPARameter:NCAScading:AGGRessor[n]:AMPLitude?
```

Related Commands RADar:SPARameter:MODE

Arguments [n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<amplitude>::= <NR3> value.

Returns A single <NR3> value.

Examples `RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:AMPLITUDE 200E-3` sets the 1st Aggressor's amplitude to 200 mV, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:AMPLITUDE?` might return `100.0000000000E-3`, indicating that the 2nd Aggressor's amplitude is set to 100 mV, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:CTALk

This command sets or returns the specified Aggressor's crosstalk type, in Non-Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:CTALk
{NEXT|FEXT|BOTH}
RADar:SPARameter:NCAScading:AGGRessor[n]:CTALk?
```

Related Commands

RADar:SPARameter:MODE

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

NEXT – Near-End Crosstalk

FEXT – Far-End Crosstalk

BOTH – Near and Far-End Crosstalk

Returns

NEXT
FEXT
BOTH

Examples

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:CTALK FEXT` sets the 1st Aggressor's Crosstalk type to Far End Crosstalk, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:CTALK?` might return `NEXT`, indicating that the 2nd Aggressor crosstalk type is set to Near End Crosstalk, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:DRATe

This command sets or returns the specified Aggressor's data rate, in Non-Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:DRATe <data_rate>
RADar:SPARameter:NCAScading:AGGRessor[n]:DRATe?
```

Related Commands RADar:SPARameter:MODE

Arguments [n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<data_rate>::= <NR3> value.

Returns A single <NR3> value.

Examples `RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:DRATE 4E9` sets the 1st Aggressor's data rate to 4 Gbps, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:DRATE?` might return `500.0000000000E+3`, indicating that the 2nd Aggressor's data rate is set to 500 kbps, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal

This command sets or returns specified Aggressor's signal type, in Non-Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal
{CLOCk|PRBS|FILE|SAVictim}
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:AGGRessor[n]:SIGNal:FILE

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

CLOCk – Indicates that the aggressor signal is a clock pattern.

PRBS – Indicates that the aggressor signal is a PBRS pattern. You also must set the PBRS type.

FILE – Aggressor is set to use a file. You must set the file path.

SAVictim – Aggressor is the same as the victim.

Returns

CLOC
PRBS
FILE
SAV

Examples

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:SIGNAL SAVICTIM` sets the 1st aggressor signal to be the same as the victim, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:SIGNAL?` might return FILE, indicating that 2nd Aggressor has a signal type set to use a file, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:FILE

This command sets or returns the filepath to the aggressor file for the specified Aggressor, in Non-Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Aggressor signal type must be File.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:FILE
<filepath>
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:FILE?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

<filepath>::= <string> defines the path to the aggressor file.

Returns

A single <filepath> string.

Examples

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:SIGNAL:FILE "C:\temp\myFile.txt"` sets the 1st Aggressor's file and filepath when the aggressor is set to use a file, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:SIGNAL:FILE?` might return `"C:\temp\myFile.txt"` indicating that the 2nd Aggressor has a signal type filepath set to "C:\temp\myFile.txt", in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:PRBS

This command sets or returns the specified Aggressor's PRBS signal type, in Non-Cascading mode for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Port configuration muse be set to 8 or 12 ports.

Aggressor signal type must be PRBS.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:PRBS
{PRBS7|PRBS9|PRBS15|PRBS16|PRBS20|PRBS21|PRBS23|PRBS29|PRBS31}
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal:PRBS?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:NCAScading:AGGRessor[n]:SIGNal

Arguments

[n] ::= {1|2} (“n” determines the aggressor signal (1 or 2). 2 is only valid if the number of ports is set to 12.)

If omitted, n is interpreted as 1.

Patterns available include: PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31

Returns

PRBS7, PRBS9, PRBS15, PRBS16, PRBS20, PRBS21, PRBS23, PRBS29, PRBS31

Examples

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR1:SIGNAL:PRBS PRBS31` sets the 1st Aggressor's Signal type's PRBS value to PRBS31, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:AGGRESSOR2:SIGNAL:PRBS?` might return `PRBS15`, indicating that the 2nd Aggressor has a signal type PRBS value set to PRBS15, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:DEEMbed

This command sets or returns whether the Non-Cascading S-Parameters is to de-embed (invert) the S-Parameters, in Non-Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Non-Cascading.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:DEEMbed {0|1|OFF|ON}
RADar:SPARameter:NCAScading:DEEMbed?
```

Related Commands RADar:SPARameter:MODE

Arguments OFF or 0 disables de-embedding. OFF or 0 is the default value.
ON or 1 enables de-embedding.

Returns A single <Boolean> value.

Examples `RADAR:SPARAMETER:NCASCADING:DEEMBED 1` will de-embed the S-Parameters for Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:DEEMBED?` might return `0`, indicating that S-Parameters will not be de-embedded for Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:DRX[n]

This command sets or returns the S-Parameter port assignment of the channel's specified receiver port number (Rx-Port) in Non-Cascading mode and Differential Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

S-Parameter Signalling Scheme must be set to Differential.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:DRX[n] <port_number>
RADar:SPARameter:NCAScading:DRX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:NCAScading:TYPE
RADar:SPARameter:NCAScading:RX[n]

Arguments

[n] ::= <NR1>. A variable value to define the receiver port number (Rx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4, then n = {1}
Type = 8 then n = {1 – 2}
Type = 12 then n = {1 – 3}

If omitted, n is interpreted as 1.

<port_number>::= <NR1> value. A variable value to define the S-Parameter Port assigned to the specified Rx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4, then n = {1 – 2}
Type = 8 then <port number> = {1 – 4}
Type = 12 then <port number> = {1 – 6}

Returns

A single <NR1> value.

Examples

`RADAR:SPARAMETER:NCASCADING:DTX2 4` assigns S-Parameter port 4 to channel's receiver port 2, in the Differential, Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:DTX3?` might return `6`, indicating that S-Parameter Port 6 is assigned to the channel's receiver port 3, in the Differential, Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:DTX[n]

This command sets or returns the S-Parameter port assignment of the channel's specified transmission port number (Tx-Port) in Non-Cascading mode and Differential Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

S-Parameter Signalling Scheme must be set to Differential.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:DTX[n] <port_number>
RADar:SPARameter:NCAScading:DTX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:NCAScading:STYPe
RADar:SPARameter:NCAScading:TYPE
RADar:SPARameter:NCAScading:DRX[n]

Arguments

[n] ::= <NR1> value. A variable value to define the transmission port number (Tx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4, then n = {1}
Type = 8 then n = {1 – 2}
Type = 12 then n = {1 – 3}

If omitted, n is interpreted as 1.

<port_number>::= <NR1> value. A variable value to define the S-Parameter Port assigned to the specified Tx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 4 then <port number> = {1 – 2}
Type = 8 then <port number> = {1 – 4}
Type = 12 then <port number> = {1 – 6}

Returns

A single <NR1> value.

Examples

`RADAR:SPARAMETER:NCASCADING:DTX2 4` assigns S-Parameter port 4 to channel's transmission port 2, in the Differential, Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:DTX3?` might return `6`, indicating that S-Parameter Port 6 is assigned to the channel's transmission port 3, in the Differential, Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:FILE

This command sets or returns the filepath and file name of the S-Parameter file, in Non-Cascading mode for the selected pulse train.

Conditions S-Parameter Mode must be set to Non-Cascading.

Requires an S-Parameters license.

Group S-Parameters

Syntax `RADar:SPARameter:NCAScading:FILE <filepath>`

Related Commands RADar:SPARameter:MODE

Arguments <filepath>::= <string> defines the path to the S-Parameter file.

Returns A single <filepath> string.

Examples `RADAR:SPARAMETER:NCASCADING:FILE "C:\temp\myFile.s12p"` sets the filepath to "C:\temp\myFile.s12p" for use during compilation for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:FILE?` might return `"C:\temp\myOtherFile.s6p"`, indicating the current filepath for the selected pulse train.

RADar:SPARameter:NCAScading:LAYout

This command sets or returns the 4-port differential signalling scheme for the S-Parameter Matrix Configuration, in Non-Cascading mode, for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Number of Ports must be set to 4 and Signalling Scheme set to Differential.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

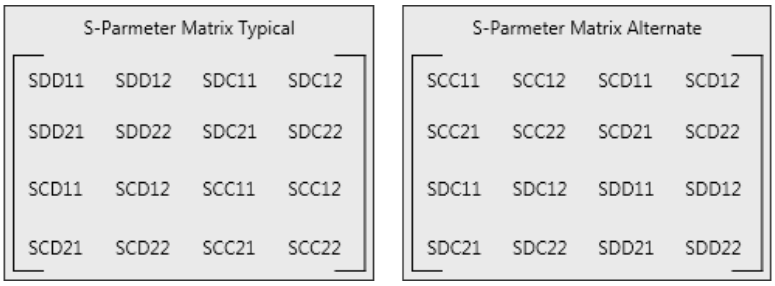
```
RADar:SPARameter:NCAScading:LAYout {TYPical|ALTernate}
RADar:SPARameter:NCAScading:LAYout?
```

Related Commands

RADar:SPARameter:MODE,
RADar:SPARameter:NCAScading:SSCHeme

Arguments

TYPical or ALTernate: selects the S-Parameter Matrix.



Returns

TYP
ALT

Examples

`RADAR:SPARAMETER:NCASCADING:LAYOUT TYPICAL` sets the 4 port configuration's Layout to Typical, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:LAYOUT?` might return `TYP`, indicating that configuration's Layout for port 4 is set to Typical, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:RX[n]

This command sets or returns the S-Parameter port assignment of the channel's specified receiver port number (Rx-Port) in Non-Cascading mode and Single-Ended Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

S-Parameter Signalling Scheme must be set to Single-Ended.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:RX[n] <port number>
RADar:SPARameter:NCAScading:RX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:NCAScading:TYPE
RADar:SPARameter:NCAScading:TX[n]

Arguments

[n] ::= <NR1>. A variable value to define the receiver port number (Rx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then n = no value
Type = 2, then n = {1}
Type = 4 then n = {1 – 2}
Type = 6 then n = {1 – 3}
Type = 8 then n = {1 – 4}
Type = 12 then n = {1 – 6}

If omitted, n is interpreted as 1.

<port number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Rx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then <port number> = no value
Type = 2, then <port number> = {1 – 2}
Type = 4 then <port number> = {1 – 4}
Type = 6 then <port number> = {1 – 6}
Type = 8 then <port number> = {1 – 8}
Type = 12 then <port number> = {1 – 12}

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:NCASCADING:RX2 4` assigns S-Parameter port 4 to the channel's receiver port 2, in the Single-Ended, Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:RX4?` might return `6`, indicating that S-Parameter Port 6 is assigned to the channel's receiver port 4, in the Single-Ended, Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:SSCHeme

This command sets or returns the S-Parameter Signalling Scheme (single-ended or differential), in Non-Cascading mode. Signalling Scheme is only available for 4, 8, or 12 port configurations for the selected pulse train.

Conditions S-Parameter Mode must be set to Non-Cascading.

Port configuration must be 4, 8, or 12 ports.

Requires an S-Parameters license.

Group S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:SSCHeme {SENDed|DIFFerential}
RADar:SPARameter:NCAScading:SSCHeme?
```

Related Commands RADar:SPARameter:MODE,
RADar:SPARameter:NCAScading:TYPE

Arguments SENDed – Single Ended Signal Scheme

DIFFerential – Differential Signal Scheme

Returns SEND
DIFF

Examples `RADAR:SPARAMETER:NCASCADING:SSCHEME DIFF` sets the Signalling Scheme to Differential, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:SSCHEME?` might return `SEND`, indicating that the Signalling Scheme is set to Single Ended, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:STYPe

This command sets or returns S-Parameter signal type (victim or aggressor), in Non-Cascading mode for the selected pulse train. The number of ports must be either 8 or 12.

Conditions

S-Parameter Mode must be set to Non-Cascading.

Port configuration must be 8 or 12 ports.

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:STYPe {VICTim|AGGRessor|BOTH}
RADar:SPARameter:NCAScading:STYPe?
```

Related Commands

RADar:SPARameter:MODE

Arguments

VICTim – enables the victim signal type.

AGGRessor – enables the aggressor signal type.

BOTH – enables the victim and aggressor signal types.

Returns

VICT
AGGR
BOTH

Examples

`RADAR:SPARAMETER:NCASCADING:STYPE BOTH` sets the signal type to include both the Victim and Aggressor signal types, in Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:STYPE?` might return `AGGR`, indicating that the S-Parameter signal type is currently set to be Aggressor, in Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:TX[n]

This command sets or returns the S-Parameter port assignment of the channel's specified transmission port number (Tx-Port) in Non-Cascading mode and Single-Ended Signalling Scheme (where applicable) for the selected pulse train.

Conditions

S-Parameter Mode must be set to Non-Cascading.

S-Parameter Signalling Scheme must be set to Single-Ended (where applicable).

Requires an S-Parameters license.

Group

S-Parameters

Syntax

```
RADar:SPARameter:NCAScading:TX[n] <port number>
RADar:SPARameter:NCAScading:TX[n]?
```

Related Commands

RADar:SPARameter:MODE
RADar:SPARameter:CASCading:STYPe
RADar:SPARameter:NCAScading:TYPE
RADar:SPARameter:NCAScading:RX[n]

Arguments

[n] ::= <NR1>. A variable value to define the transmission port number (Tx-Port) of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then n = no value
Type = 2, then n = {1}
Type = 4 then n = {1 – 2}
Type = 6 then n = {1 – 3}
Type = 8 then n = {1 – 4}
Type = 12 then n = {1 – 6}

If omitted, n is interpreted as 1.

<port number> ::= <NR1>. A variable value to define the S-Parameter Port assigned to the specified Tx-Port of the channel.

The actual range is dependent on the Number of Ports (Type).

Type = 1, then <port number> = no value
Type = 2, then <port number> = {1 – 2}
Type = 4 then <port number> = {1 – 4}
Type = 6 then <port number> = {1 – 6}
Type = 8 then <port number> = {1 – 8}
Type = 12 then <port number> = {1 – 12}

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:NCASCADING:TX2 4` assigns S-Parameter port 4 to the channel's transmission port 2, in the Single-Ended, Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:TX4?` might return `6`, indicating that S-Parameter Port 6 is assigned to the channel's transmission port 4, in the Single-Ended, Non-Cascading mode for the selected pulse train.

RADar:SPARameter:NCAScading:TYPE

This command sets or returns the S-Parameter number of ports, in Non-Cascading mode for the selected pulse train.

Conditions Requires an S-Parameters license.

Group S-Parameters

Syntax `RADar:SPARameter:NCAScading:TYPE {1|2|4|6|8|12}`

Arguments {1|2|4|6|8|12} – defines the number of S-Parameter ports.

Returns A single <NR1> value.

Examples `RADAR:SPARAMETER:NCASCADING:TYPE 12` sets the S-Parameter type to a 12-Port system for Non-Cascading mode for the selected pulse train.

`RADAR:SPARAMETER:NCASCADING:TYPE?` might return `6`, indicating that the S-Parameter type is a 6-Port system for Non-Cascading mode for the selected pulse train.

WPLugin:ACTive

This command sets or returns the active waveform plug-in.

To use the Radar Signal commands in this document, the active waveform plug-in must be set to Radar.

Group Control

Syntax

```
WPLugin:ACTive <plug-in_name>
WPLugin:ACTive?
```

Arguments <plug-in_name>::=<string>

A single string representing the waveform plug-in name.

`"Radar"` is the proper string to activate the Radar signal plug-in.

Returns <plug-in_name>::=<string>

Examples `WPLUGIN:ACTIVE "Radar"` sets the Radar Signal plug-in as the active plug-in.

`WPLUGIN:ACTIVE?` might return `"Radar"`, indicating Radar is currently the active waveform plug-in.

Index

W