Tektronix®

Series 2260B

Programmable DC Power Supplies

PROGRAMMING

077104703 October 2024

KEITHLEY
A Tektronix Company

Series 2260B

Programmable DC Power Supplies

Programming

Cleveland, Ohio, U.S.A.

These are the original instructions in English.

Document number: 077104703 October 2024

KEITHLEY
A Tektronix Company

Safety precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with nonhazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the user documentation for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product warranty may be impaired.

The types of product users are:

Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

Maintenance personnel perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

Service personnel are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Keithley products are designed for use with electrical signals that are measurement, control, and data I/O connections, with low transient overvoltages, and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II (as referenced in IEC 60664) connections require protection for high transient overvoltages often associated with local AC mains connections. Certain Keithley measuring instruments may be connected to mains. These instruments will be marked as category II or higher.

Unless explicitly allowed in the specifications, operating manual, and instrument labels, do not connect any instrument to mains.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

For safety, instruments and accessories must be used in accordance with the operating instructions. If the instruments or accessories are used in a manner not specified in the operating instructions, the protection provided by the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories. Maximum signal levels are defined in the specifications and operating information and shown on the instrument panels, test fixture panels, and switching cards.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as protective earth (safety ground) connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

If a  screw is present, connect it to protective earth (safety ground) using the wire recommended in the user documentation.

The  symbol on an instrument means caution, risk of hazard. The user must refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

The  symbol on an instrument means warning, risk of electric shock. Use standard safety precautions to avoid personal contact with these voltages.

The  symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The  symbol indicates a connection terminal to the equipment frame.

If this  symbol is on a product, it indicates that mercury is present in the display lamp. Please note that the lamp must be properly disposed of according to federal, state, and local laws.

The **WARNING** heading in the user documentation explains hazards that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in the user documentation explains hazards that could damage the instrument. Such damage may invalidate the warranty.

The **CAUTION** heading with the  symbol in the user documentation explains hazards that could result in moderate or minor injury or damage the instrument. Always read the associated information very carefully before performing the indicated procedure. Damage to the instrument may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits — including the power transformer, test leads, and input jacks — must be purchased from Keithley. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. The detachable mains power cord provided with the instrument may only be replaced with a similarly rated power cord. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keithley to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley office for information.

Unless otherwise noted in product-specific literature, Keithley instruments are designed to operate indoors only, in the following environment: Altitude at or below 2,000 m (6,562 ft); temperature 0 °C to 50 °C (32 °F to 122 °F); and pollution degree 1 or 2.

To clean an instrument, use a cloth dampened with deionized water or mild, water-based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., a data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

Safety precaution revision as of June 2018.

Table of contents

Section 1

Introduction

In this section:

Features

The key features include:

- 360 W, 720 W, and 1080 W versions with voltage up to 800 V and currents up to 108 A.
- Programmable voltage or current rise and fall slew rates prevent damage to low impedance loads from inrush current.
- Constant current priority setting reduces voltage and current overshoot when powering LEDs.
- Programmable output resistance allows simulation of the output characteristics of a battery.
- Analog input control supports creating voltage profiles that simulate how a battery responds to a fast changing load current or generating a combined DC output level with an AC signal superimposed on it to test how a circuit responds to noise on its DC power source.
- Internal test sequence mode allows the output of a set of voltage levels for varying time intervals at each level. The test sequence can be stored and used repeatedly to determine DUT performance at different levels.
- Series or parallel configurable, which can deliver up to 3240 W when three units are combined in parallel.
- Fit six 360 W instrument, three 720 W instruments, or two 1080 W instruments in a standard rack width to minimize bench and test system space.
- Fast discharge capability and 1 ms transient recovery time to load changes minimizes test times in automated test systems.

2260B series overview

The 2260B series includes power supplies with 360 W, 720 W, and 1080 W power capabilities.

In this document:

- 2260B-30 refers to 2260B models with a maximum voltage rating of 30 V
- 2260B-80 refers to models with a maximum voltage rating of 80 V
- 2260B-250 refers to models with a maximum voltage rating of 250 V
- 2260B-800 refers to models with a maximum voltage rating of 800 V

This document provides information on the 2260B models listed in the following table.

Model name	Type	Voltage rating	Current rating	Power
2260B-30-36	360 W models	0 V to 30 V	0 A to 36 A	360 W
2260B-80-13	360 W models	0 V to 80 V	0 A to 13.5 A	360 W
2260B-250-4	360 W models	0 V to 250 V	0 A to 4.5 A	360 W
2260B-800-1	360 W models	0 V to 800 V	0 A to 1.44 A	360 W
2260B-30-72	720 W models	0 V to 30 V	0 A to 72 A	720 W
2260B-80-27	720 W models	0 V to 80 V	0 A to 27 A	720 W
2260B-250-9	720 W models	0 V to 250 V	0 A to 9 A	720 W
2260B-800-2	720 W models	0 V to 800 V	0 A to 2.88 A	720 W
2260B-30-108	1080 W models	0 V to 30 V	0 A to 108 A	1080 W
2260B-80-40	1080 W models	0 V to 80 V	0 A to 40.5 A	1080 W
2260B-250-13	1080 W models	0 V to 250 V	0 A to 13.5 A	1080 W
2260B-800-4	1080 W models	0 V to 800 V	0 A to 4.32 A	1080 W

Each power supply differs in size. The 720 W and 1080 W models are larger than the 360 W models to accommodate the increase in power.

Figure 1: 360 W models

Figure 2: 720 W models

Figure 3: 1080 W models

Extended warranty

Additional years of warranty coverage are available on many products. These valuable contracts protect you from unbudgeted service expenses and provide additional years of protection at a fraction of the price of a repair. Extended warranties are available on new and existing products. Contact your local Keithley Instruments office, sales partner, or distributor for details.

Contact information

If you have any questions after you review the information in this documentation, please contact your local Keithley Instruments office, sales partner, or distributor. You can also call the Tektronix corporate headquarters (toll-free inside the U.S. and Canada only) at 1-800-833-9200. For worldwide contact numbers, visit tek.com/contact-tek.

Section 2

Remote operations

In this section:

Remote communications

You can control the 2260B remotely, as described in the following topics. For a command list, refer to SCPI command reference (on page 4-1).

The 2260B supports the following remote interfaces:

- **GPIB:** IEEE-488 instrumentation general-purpose interface bus
- **Ethernet:** Local-area-network communications
- **USB:** Type B USB port

The 2260B can be controlled from only one communications interface at a time.

USB

To communicate from a computer to the instrument, you need a USB cable with a USB Type B connector end and a USB Type A connector end.

USB speed is 1.1/2.0 (full speed/high speed). The USB class is CDC (communications device class).

To set up the 2260B for use with USB communications:

1. Turn off the 2260B.
2. Connect the USB cable Type A connector to the computer.
3. Connect the USB cable Type B connector to the rear-panel USB port.
4. Turn the 2260B on.
5. Press the **Function** key.
6. Select **F-22**.
7. Select **2** (USB-CDC, Auto Detect Speed).

Check USB functionality

To check functionality of the USB, you can use the NI™ Measurement and Automation Explorer (NI MAX). This program is available on the NI website, ni.com. The 2260B must be firmware version 1.12 or above to use NI MAX. You can use :SYSTem:INFormation? (on page 4-38) to check the firmware version number.

NOTE

The 2260B appears as a COM port on the computer. Use the Device Manager to find the COM port number. The displayed COM port number may differ from the following description, depending on the virtual COM port number in your system.

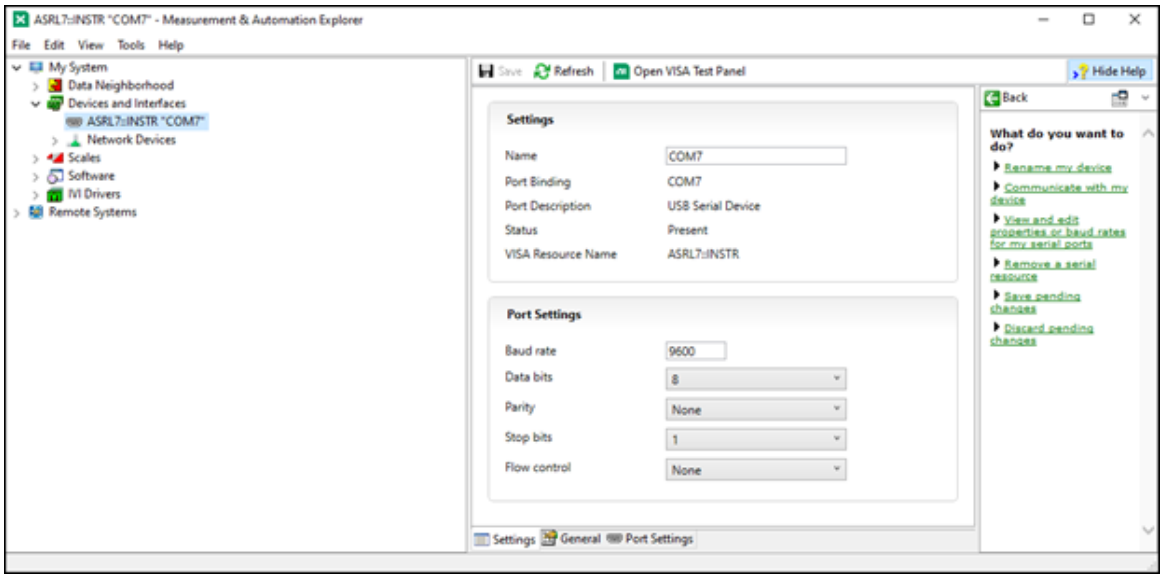
To check USB functionality:

1. To start NI MAX, type `NI MAX` in the Windows **Start** menu.
2. From the Configuration Panel, select **My System > Device and Interfaces**, as shown in the following figure.

Figure 4: Device and interface check

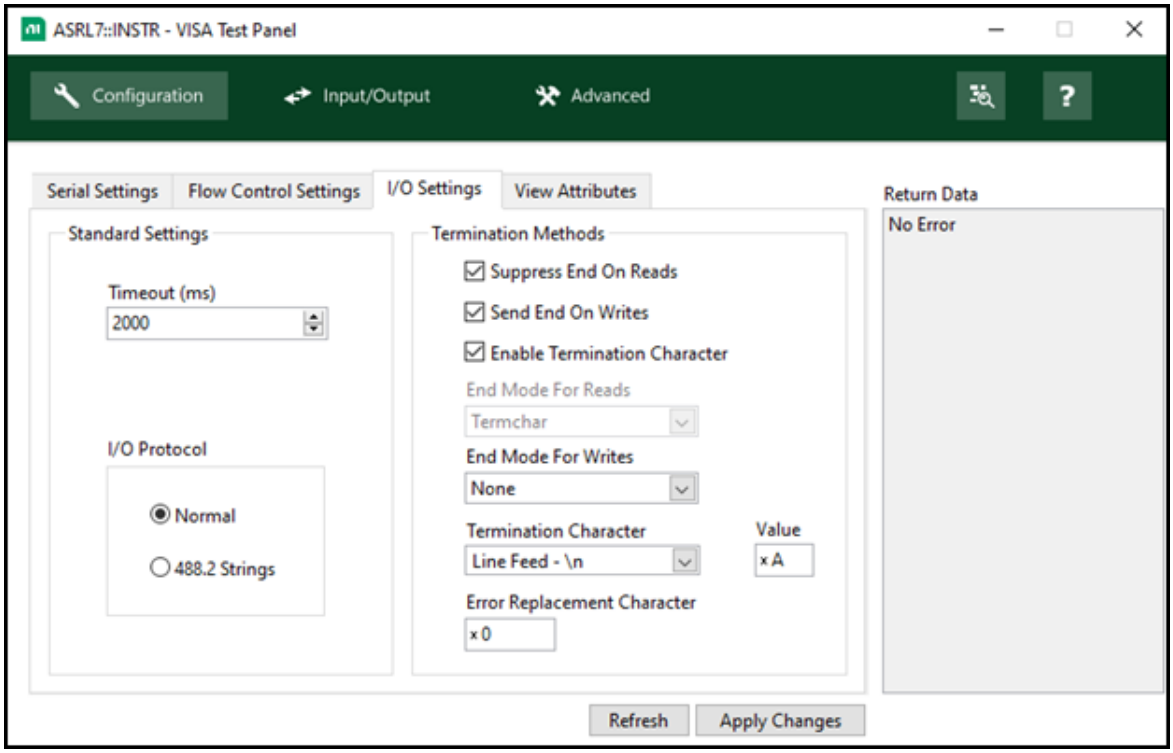
3. Select **COM7**.
4. Select **Open VISA Test Panel**.

Figure 5: Open VISA Test Panel



5. Select **Configuration**.

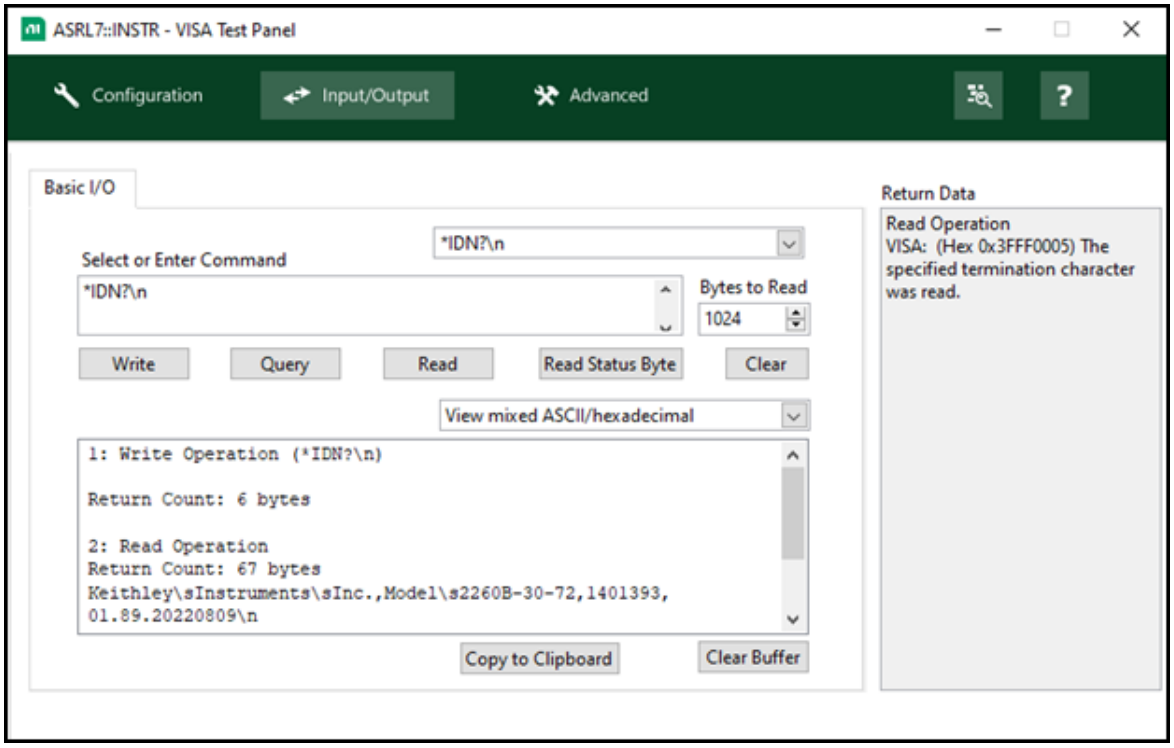
Figure 6: Configuration I/O Settings screen



6. In the I/O Settings tab, select **Enable Termination Character**.
7. Set the Termination Character to **Line Feed - \n**.

8. Select **Apply Changes**.
9. Select **Input/Output**.

Figure 7: NI MAX Input/Output settings



10. In **Select or Enter Command**, ensure that ***IDN?\n** is configured.
11. Select **Query**. The `*IDN?` query is returned to the buffer area as shown in the previous figure.

Ethernet

You can configure the ethernet interface for any of the following applications:

- Basic remote control
- Monitoring using a web server
- Socket server

The 2260B series supports DHCP connections so the instrument can be automatically connected to an existing network. You can also manually configure the network settings. For detail on the available settings, refer to LAN settings.

The ethernet configuration parameters are:

- MAC Address (display only)
- LAN
- DHCP
- IP Address
- Subnet Mask
- Gateway
- DNS Address
- Sockets Active
- Web Server Active
- Web Password Active
- Web set password: 0000 to 9999 (default 0000)

NOTE

Contact your network administrator to confirm your specific network requirements before setting up a LAN connection.

Web server configuration

This configuration example configures the 2260B as a web server and uses DHCP to automatically assign an IP address to the 2260B.

To set up the 2260B as a web server:

1. Connect an ethernet cable from the LAN network to the rear-panel ethernet port.
2. Press the **Function** key.
3. Select **F-36** (LAN enable or disable).
4. Select **1** (Enable).
5. Select **F-37** (DHCP enable or disable).
6. Select **1** (Enable).
7. Select **F-59** (Web server control enable or disable)
8. Select **1** (Enable).
9. To verify web server functionality, enter the IP address of the power supply into a web browser. The web browser interface is displayed. You can use F-39, F-40, F-41, and F-42 to retrieve the IP address.

NOTE

You may need to cycle the power or refresh the web browser to connect to a network.

Socket server configuration

This configuration example demonstrates how to configure the 2260B socket server.

In the following procedure, you manually assign an IP address to the 2260B and enable the socket server. By default, the socket server port number is 2268 and cannot be changed.

NOTE

The socket function is only available for firmware version V1.12 or above. You can use :SYSTem:INFormation? (on page 4-38) to check your firmware version number.

To set up the 2260B as a socket server:

1. Connect an ethernet cable from the LAN network to the rear-panel ethernet port.
2. Press the **Function** key.
3. Select **F-36** (LAN enable or disable).
4. Select **1** (Enable).
5. Select **F-37** (DHCP enable or disable).
6. Select **0** (Disable).
7. Set the IP address using **F-39**, **F-40**, **F-41**, and **F-42**.
8. Set the subnet mask using **F-43**, **F-44**, **F-45**, and **F-46**.
9. Set the gateway using **F-47**, **F-48**, **F-49**, and **F-50**.
10. Select **F-57** (WebSocket enable or disable).
11. Select **1** (Enable).

NOTE

You may need to cycle the power or refresh the web browser to connect to a network.

Socket server function check

To test the socket server functionality, you can use the NI™ Measurement and Automation Explorer (NI MAX). This program is available on the NI website, ni.com. The 2260B must be firmware version 1.12 or above to use NI MAX. You can use :SYSTem:INFormation? (on page 4-38) to check the firmware version number.

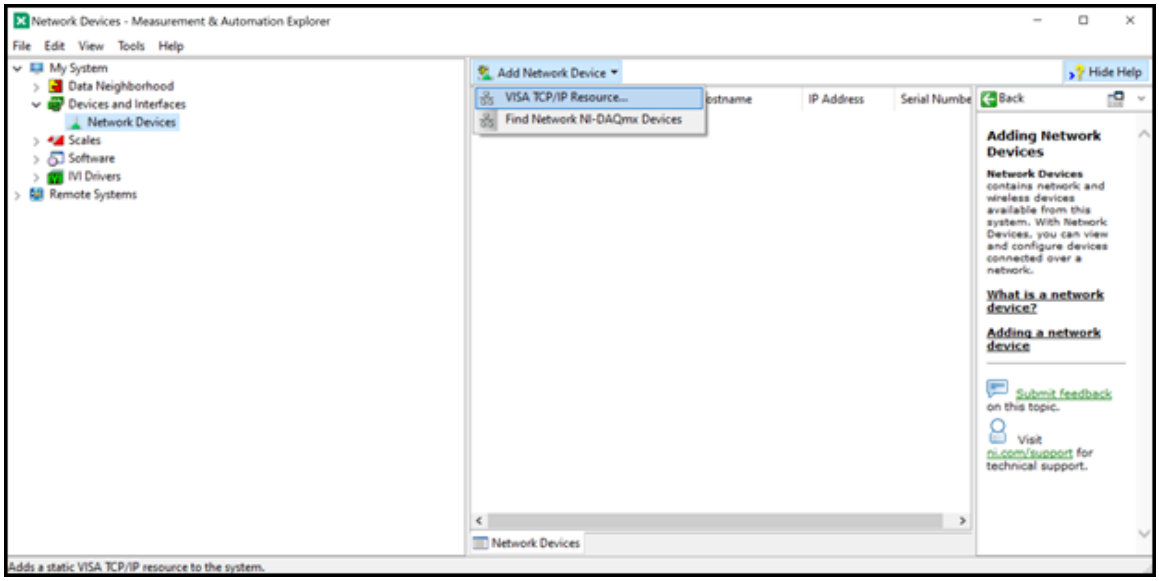
To check socket server functionality:

1. To start NI MAX, type `NI MAX` in the Windows **Start** menu.
2. From the Configuration Panel, select **My System > Device and Interfaces**, as shown in the following figure.

Figure 8: Device and interface check

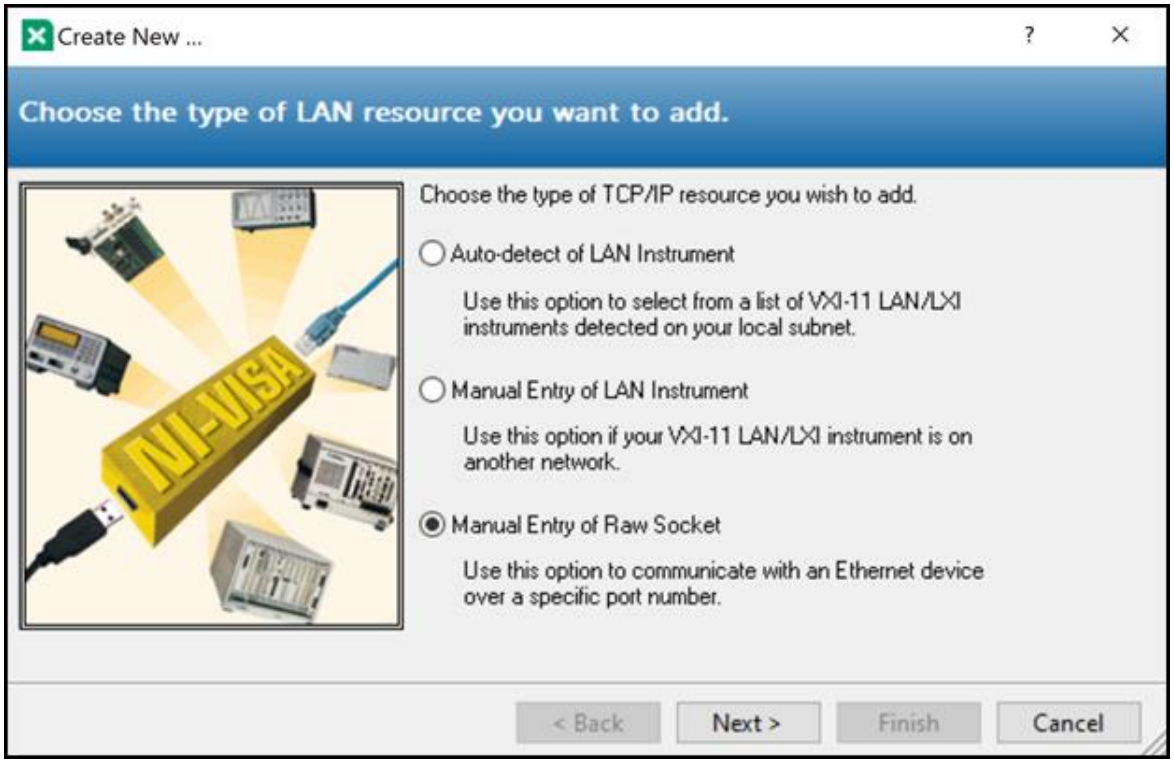
3. Select **Network Devices**.
4. Select **Add Network Device > VISA TCP/IP Resource**, as shown in the following figure.

Figure 9: Add a network device



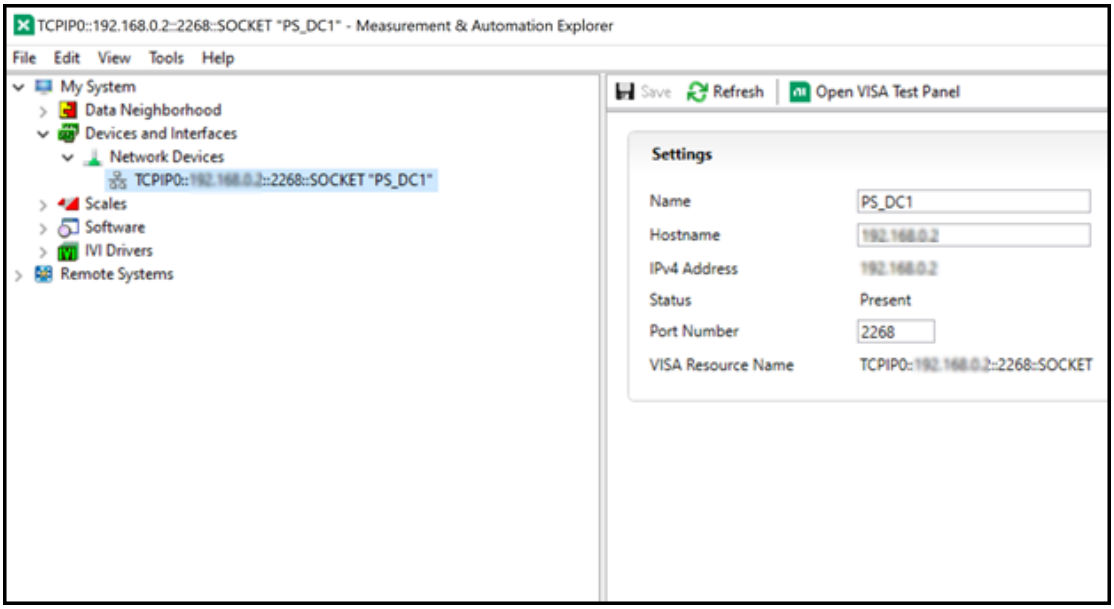
5. Select **Manual Entry of Raw Socket**.

Figure 10: Select Manual Entry of Raw Socket



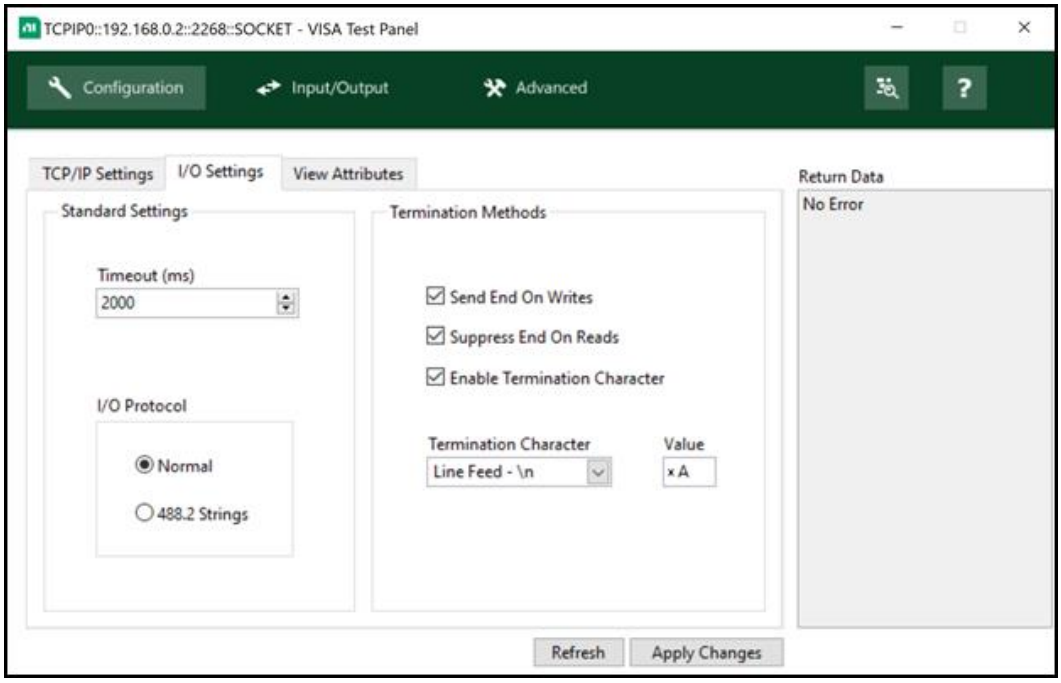
6. Select **Next**. The Enter the LAN Resource Details dialog is displayed.
7. Enter the **IP address** and the **port number** of the 2260B. The port number is fixed at 2268.
8. Select **Validate**. A message is displayed when the validation is successful.
9. Select **Next**. The Specify an Alias for this Resource dialog is displayed.
10. Enter an alias name for the 2260B connection, such as `PS_DC1`.
11. Select **Finish**. The IP address of the 2260B is shown under Network Devices in the configuration panel, as shown in the following figure.
12. Select the new network device.

Figure 11: Select network device



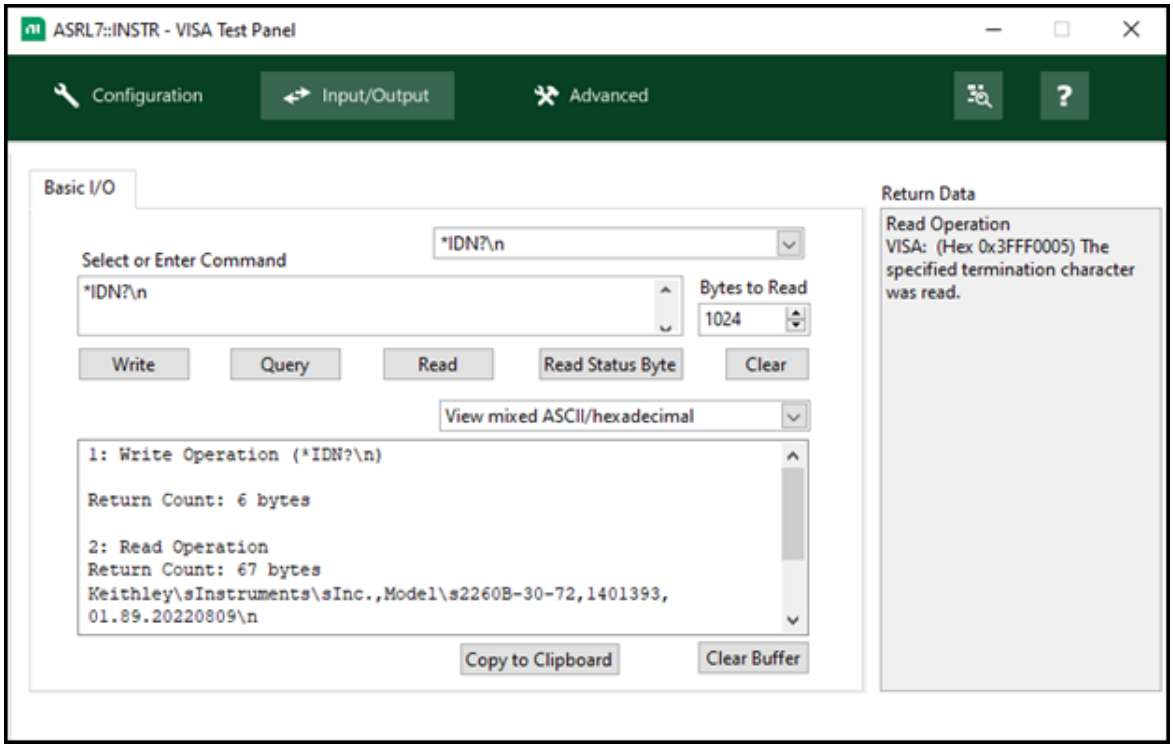
13. Select **Open VISA Test Panel**.
14. Select **Configuration**.
15. In the **I/O Settings** tab, select **Enable Termination Character**.
16. In the **Termination Character** list, ensure that **Line Feed - \n** is selected.
17. Click **Apply Changes**.

Figure 12: I/O Settings



18. Select **Input/Output**.

Figure 13: NI MAX Input/Output settings



19. In **Select or Enter Command**, make sure ***IDN?\n** is configured.
20. Select **Query**. The `*IDN?` query should be returned to the buffer area.

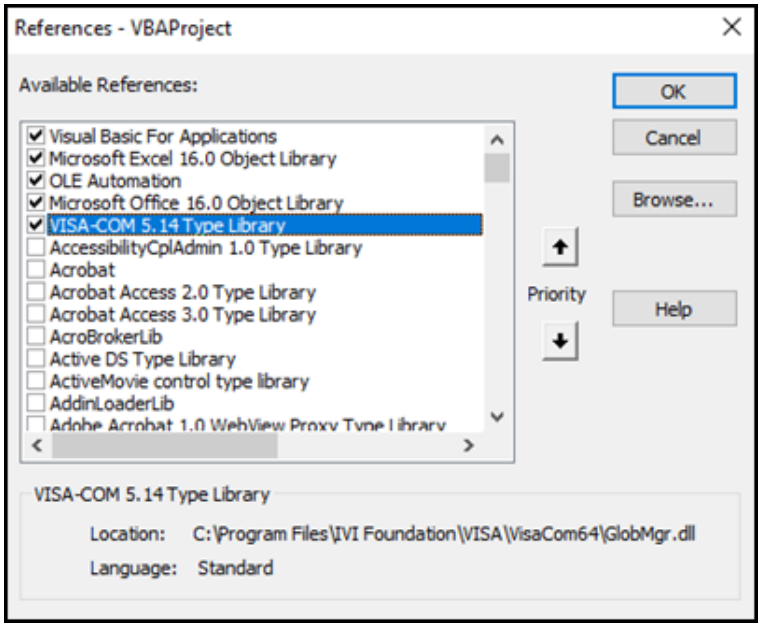
Socket server examples

The following examples demonstrate connections to the 2260B using Visual Basic and C++.

Visual Basic example

This example uses the IP address of 172.15.5.133 over port 2268 to connect to the 2260B. The program sends the `*IDN?` query to the 2260B, displays the return string in the message box, and then closes the connection. This example uses the VISA-COM 5.14 Type Library.

Figure 14: C++ Example




```
'Create VISA ResourceManager object
    Dim rm As New VisaComLib.ResourceManager
    Dim accessMode As VisaComLib.accessMode
    Dim serial As String
    Dim timeOut As Integer
    Dim optionString As String
    Dim inst2260B As VisaComLib.IMessage
    Dim inst2260Bsfc As VisaComLib.IAsyncMessage

Private Sub CommandButton1_Click()
    accessMode = VisaComLib.accessMode.NO_LOCK
    timeOut = 0
    optionString = ""

    'Connect to the 2260B
    Set inst2260B = rm.Open("TCPIP0::192.168.0.2::2268::SOCKET", accessMode,
   timeOut, optionString)
    Set inst2260Bsfc = inst2260B
    inst2260Bsfc.TerminationCharacterEnabled = True

    'Query the System Identify Name
    inst2260B.WriteString ("*IDN?" & vbLf)
    MsgBox inst2260B.ReadString(256)

    'Close communications
    inst2260B.Close

End Sub
End Sub
```

C++ example

The following program:

- Creates a connection to the 2260B
- Sets the voltage to 3.3 volts
- Sets the current to 1.5 amperes
- Reads the voltage and current readings
- Closes the connection

NOTE

Add `visa32.lib` to the project library when building the following sample program.

```
include "stdio.h"
include "string.h"
include "visatype.h"
include "visa.h"
define IPaddr "172.16.20.181"
int main(int argc, char* argv[])
{
   UiSession defaultRm, instr;
   // Create VISA ResourceManager object
   ViStatus status = viQpenDefaultRM(&defaultRm);
   if (status < VI SUCCESS)
   {
     // Initialization error
      return -1 ;
   }
   ViChar rsc[256];
   sprintf(rsc, "TCPIP0::%s::2268::SOCKET", IPaddr);
   ViAccessMode accessMode = VI_N0_LOCK;
   ViUint32 timeout = 0;
   // Connect the device
   viOpen(defaultRm, rsc, accessMode, timeout, &instr);
   /* Set the timeout for message-based communication */
   status = viSetAttribute(instr, VI_ATTR_TMO_UALUE, 5000);
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR, 10);
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_TRUE);
   ViUInt32 count;
   // Set the voltage to 3.3, current to 1.5
   ViBuf buf = (ViBuf)":volt 3.3;:curr 1.5\n";
   viWrite(instr, buf, (ViUInt32)strlen((ViPChar)buf), &count);

   // Query the voltage and current
   buf = (ViBuf)":apply?\n";
   status =viWrite(instr, buf, (ViUInt32)strlen((ViPChar)buf), &count);
   ViChar result[257];
   status =viRead(instr, (ViPBuf)result, 256, &count);
```

```
   if (status=VI_SUCCESS_TERM_CHAR)
   {
      result[count] = 0;
      printf("Voltage(V), Current(A)= %s\n", result);
   }else
      printf ("Error\n");
   // Close the device
   viClose(instr);
   viClose(defaultRm);

   return 0;
}
```

GPIB

NOTE

The 2260B-GPIB-USB accessory is no longer available for purchase. The following information is provided to support customers who have previously purchased this accessory.

You must use the 2260B-GPIB-USB accessory to use GPIB communications with the 2260B.

When using GPIB, be aware of the following constraints:

- Maximum of 15 devices, including the controller
- Maximum cable length of 20 m (65.6 ft), with 2 m (6.5 ft) between each device
- Each device must have a unique address
- At least 2/3 of the devices must be on
- No loop or parallel connections

You may see erratic bus operation if you ignore these limits.

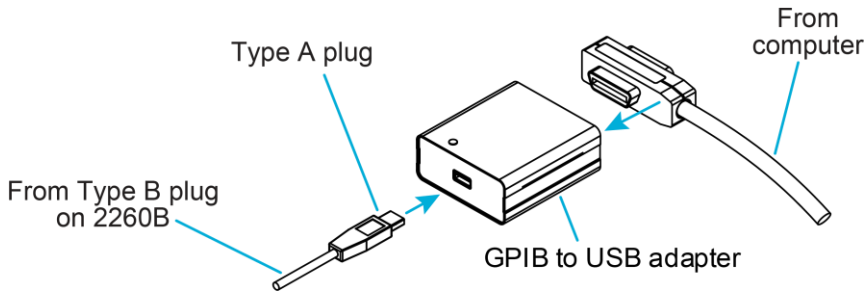
GPIB controllers are usually set to address 0 or 21. To avoid address conflicts, do not configure any instrument to have an address of 0 or 21.

Configure GPIB

To set up the 2260B for use with GPIB communications:

1. Turn off the 2260B.
2. Connect the USB cable from the rear panel USB B port on the 2260B to the USB A port on the GPIB-to-USB adapter, as shown in the following figure.

Figure 15: GPIB connections



3. Connect a GPIB cable from a GPIB controller to the GPIB port on the adapter.
4. Turn the 2260B on.
5. Press the **Function** key.
6. Select **F-22**.
7. Select **1**. This sets the rear panel USB port to GPIB-USB.
8. Select **F-23**.
9. Set the GPIB address (0 to 30).

Section 3

Introduction to SCPI commands

In this section:

Programming syntax

The Standard Commands for Programmable Instruments (SCPI) standard is a syntax and set of commands that are used to control test and measurement devices. For more information, see the IEEE-488.2 and SCPI standards.

The following information describes some basic SCPI command information and how SCPI is used with the 2260B and presented in the 2260B documentation.

Commands that are listed as SCPI compliant have commands and parameters that are SCPI confirmed. Other commands are SCPI commands, but do not conform to the SCPI standard set of commands, which means it is not a recognized command by the SCPI consortium. SCPI confirmed commands that use one or more non-SCPI parameters are explained in the “Details” section of the command description.

Command words

Program messages are made up of one or more command words.

Some command words are enclosed in brackets ([ ]). These brackets denote an optional command word that does not need to be included in the program message. For example:
`:INITiate[:IMMediate]`

These brackets indicate that `:IMMediate` is an optional command word and does not have to be used. When using optional command words in your program, do not include the brackets. For example, the above command can be sent in either of these ways:

```
:INITiate
```

```
:INITiate:IMMediate
```

Commands and command parameters

Common commands and SCPI commands may or may not use a parameter. Parameters are shown in angle brackets (< >). The following are some examples:

`*SRE <NRf>`	Parameter (`NRf`) required.
`*RST`	No parameter used.
`:OUTPut <b>`	Parameter `<b>` required.
`:SYSTem:PRESet`	No parameter used.

Put at least one space between the command word and the parameter. Do not include the angle brackets when sending the command.

Parameter types

The following table lists the common parameter types.

`<b>`	Boolean. Enables or disables an instrument operation. `0` or `OFF` disables the operation, and `1` or `ON` enables the operation. For example, the following command enables the beeper of the 2260B: `:SYSTem:CONFigure:BEEPer ON`
`<name>`	Name parameter. Select a parameter name from a listed group. For example, to set the output mode to be CV high speed priority: `:OUTPut:MODE CVHS`
`<NRf>`	Numeric representation format. A number that can be expressed as an integer (such as 8), a real number (such as 23.6), or an exponent (2.3e6). For example, to set the rising current slew rate to 72 A/s: `:SOURce:CURRent:SLEW:RISing 72`

`<n>`	Numeric value. Can consist of an NRf number or one of the following name parameters: `DEFault`, `MINimum`, or `MAXimum`. When the `DEFault` parameter is used, the instrument is programmed to the `*RST` default value. When the `MINimum` parameter is used, the instrument is programmed to the lowest allowable value. When the `MAXimum` parameter is used, the instrument is programmed to the largest allowable value.
	For example, to set the overcurrent protection level to 4 A:
	`:SOURce:CURRent:PROTection 4`
	To set the overcurrent protection level to the minimum value:
	`:SOURce:CURRent:PROTection MIN`

Long-form and short-form versions

This documentation shows SCPI commands with both uppercase and lowercase letters. The uppercase letters are the required elements of a command. The lowercase letters are optional. If you choose to include the letters that are shown in lowercase letters, you must include all of them.

When you send a command to the instrument, letter case is not important. You can mix uppercase and lowercase letters in program messages.

For example, you can send the command `:SYSTem:PRESet` in any of the following formats:

```
:SYSTem:PRESet
:SYST:PRES
:SYSTem:PRES
:syst:pres
```

Query commands

SCPI queries have a question mark (`?`) after the command. You can use the query to determine the present value of the parameters of the command or to get information from the instrument.

For example, to determine the present setting for the overcurrent protection, you can send:

```
:CURRent:PROTtection?
```

This query returns the present setting.

If the command has `MINimum`, `MAXimum`, and `DEFault` options, you can use the query command to determine what the minimum, maximum, and default values are. In these queries, the `?` is placed before the `MINimum`, `MAXimum`, or `DEFault` parameter. For example, to determine the default value for the timer, you can send:

```
:CURRent:PROTtection? MINimum
```

If you send two query commands without reading the response from the first, and then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query command without reading the response. When you cannot avoid this situation, send a device clear before sending the second query command.

Program messages

A program message consists of one or more command words sent by the computer to the instrument.

Each common command is a three-letter acronym preceded by an asterisk (`*`). Common commands are described in Common commands (on page 5-1).

SCPI commands consist of several levels. The following discussion uses the `:STATus` subsystem to explain how command words are structured to create program messages.

Command structure

`:STATus`	Path (root)
`:OPERation`	Path
`:ENABle <NRf>`	Command and parameter
`:ENABle?`	Query command
`:PRESet`	Command

Single command messages

The `:STATus` command structure has three levels. The first level is made up of the root command (`:STATus`) and serves as a path. The second level is made up of another path (`:OPERation`) and a command (`:PRESet`). The third level is made up of one command for the `:OPERation` path. You can execute these commands by sending the following program messages:

```
:STAT:OPER:ENAB  1
:STAT:OPER:ENAB?
:STAT:PRES
```

In each of these program messages, the path pointer starts at the root command (`:STAT`) and moves down the command levels until the command is executed.

Multiple command messages

You can send multiple command messages in the same program message if they are separated by semicolons (;). The following is an example showing two commands in one program message:

```
:STAT:OPER; :STAT:OPER:ENAB 1
```

When this command is sent, the first command word is recognized as the root command (`:STAT`). When the next colon is detected, the path pointer moves to the next command level and executes the command. When the path pointer sees the colon after the semicolon (`;`), it resets to the root level.

Commands that are on the same command level can be executed without having to retype the entire command path. For example:

```
:STAT:OPER:ENAB 1; ENAB?
```

After the first command (`:ENAB`) is executed, the path pointer is at the third command level in the structure.

Command path rules

Each new program message must begin with the root command unless it is optional, such as `[:SENSe]`. If the root is optional, treat the command word on the next level as the root.

The colon (`:`) at the beginning of a program message is optional.

The path pointer can only move down. It cannot be moved up a level. Executing a command at a higher level requires that you start over at the root command.

Using common commands and SCPI commands in the same message

You can use common commands and SCPI commands in the same message if they are separated by semicolons (`;`). A common command can be executed at any command level and does not affect the path pointer.

```
:STAT:OPER:ENAB 1; *ESE 1
```

Program message terminator (PMT)

Each program message must be terminated with a line feed (LF), end or identify (EOI), or LF+EOI. The bus hangs if your computer does not provide this termination. The following example shows how a program message must be terminated:

```
:ROUT:SCAN  (@1:5)  <PMT>
```

Command execution rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and is not executed.
- Valid commands that precede an invalid command in a multiple command program message are executed.
- Valid commands that follow an invalid command in a multiple command program message are ignored.

Sending strings

If you are sending a string, it must begin and end with matching quotes (either single quotes or double quotes). To include a quote character as part of the string, type it twice with no characters in between.

Response messages

A response message is the message sent by the instrument to the computer in response to a query command program message.

After sending a query command, the response message is placed in the Output Queue. When the 2260B is addressed to talk, the response message is sent from the Output Queue to the computer.

Each response is terminated with a line feed (LF) and end or identify (EOI). The following example shows how a multiple response message is terminated:

```
0; 1; 1; 0;  <RMT>
```

Multiple response messages

If you send more than one query command in the same program message, the response messages for all the queries are sent to the computer when the 2260B is addressed to talk. The responses are sent in the order that the query commands were sent and are separated by semicolons (;). Items in the same query are separated by commas (,). The following example shows the response message for a program message that contains four single item query commands:

```
0; 1; 1; 0
```

Message exchange protocol

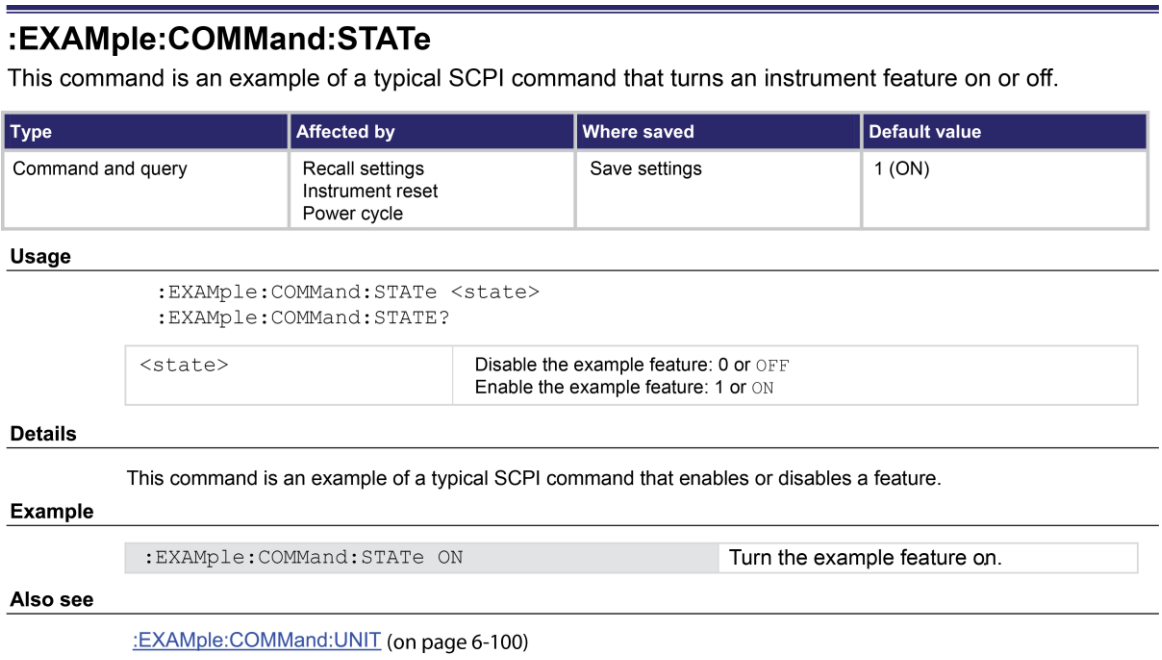
These rules summarize the message exchange protocol:

1. Always tell the 2260B what to send to the computer. To send information from the instrument to the computer:
    a. Send the appropriate query commands in a program message.
    b. Address the 2260B to talk.
2. The complete response message must be received by the computer before another program message can be sent to the 2260B.

Using the SCPI command reference

The SCPI command reference contains detailed descriptions of each of the SCPI commands that you can use to control your instrument. Each command description is broken into several standard subsections. The following figure shows an example of a command description.

Figure 16: SCPI command description example



:EXAMple:COMMand:STATe

This command is an example of a typical SCPI command that turns an instrument feature on or off.

Type	Affected by	Where saved	Default value
Command and query	Recall settings Instrument reset Power cycle	Save settings	1 (ON)

Usage

```
:EXAMple:COMMand:STATe <state>
:EXAMple:COMMand:STATE?
```

<state>	Disable the example feature: 0 or OFF Enable the example feature: 1 or ON

Details

This command is an example of a typical SCPI command that enables or disables a feature.

Example

:EXAMple:COMMand:STATe ON	Turn the example feature on.

Also see

:EXAMple:COMMand:UNIT (on page 6-100)

Each command listing is divided into five subsections that contain information about the command:

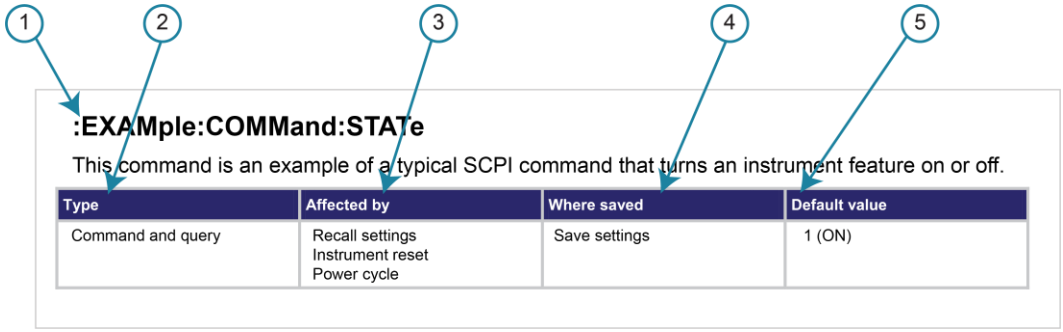
- Command name and summary table
- Usage
- Details
- Example
- Also see

The content of each of these subsections is described in the following topics.

Command name and summary table

Each instrument command description starts with the command name, followed by a table with relevant information for each command. Definitions for the numbered items are listed following the figure.

Figure 17: SCPI command name and summary table



1 **Instrument command name**. Signals the beginning of the command description and is followed by a brief description of what the command does.

2 **Type of command**. Options are:

- **Command only**. There is a command but no query option for this command.
- **Command and query**. The command has both a command and query form.
- **Query only**. This command is a query.

3 **Affected by**. Commands or actions that have a direct effect on the instrument command.

- **Recall settings.** If you send `*RCL` to recall the system settings, this setting is changed to the saved value.
- **Instrument reset.** When you reset the instrument, this command is reset to its default value. Reset can be done from the front panel or when you send `*RST`.
- **Power cycle.** When you power cycle the instrument, this command is reset to its default value.

4 **Where saved**. Indicates where the command settings reside once they are used on an instrument. Options include:

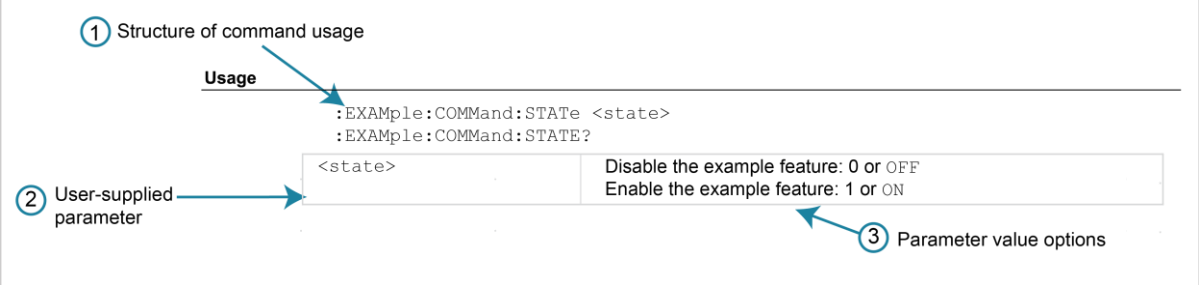
- **Not saved.** Command is not saved and must be sent each time you use it.
- **Nonvolatile memory.** The command is stored in a storage area in the instrument where information is saved even when the instrument is turned off.
- **Save settings.** This command is saved when you send the `*SAV` command.

5 **Default value:** Lists the default value for the command. The parameter values are defined in the Usage or Details sections of the command description.

Command usage

The Usage section of the remote command listing shows how to properly structure the command. Each line in the Usage section is a separate variation of the command usage; all possible command usage options are shown here.

Figure 18: SCPI command description usage identification

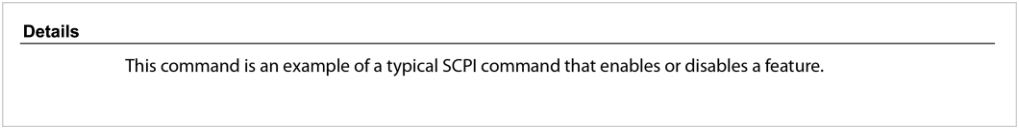


1. **Structure of command usage:** Shows the organization of the parts of the command.
2. **User-supplied parameters:** Indicated by angle brackets (< >).
3. **Parameter value options:** Descriptions of the options that are available for the parameter.

Command details

This section lists additional information you need to know to successfully use the command.

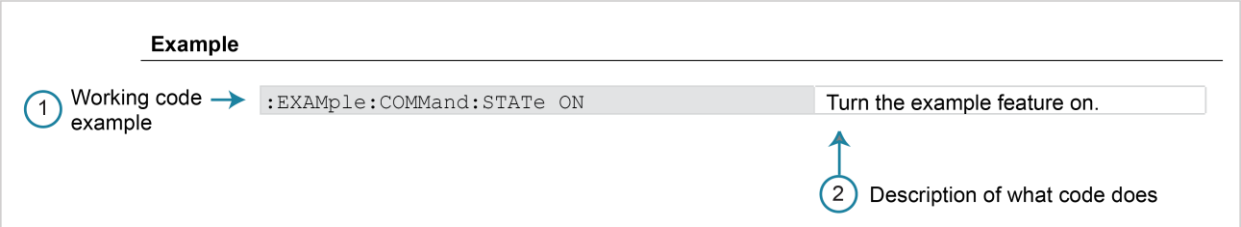
Figure 19: Details section of command listing



Example section

The Example sections of the command descriptions show simple examples of how the commands can be used.

Figure 20: SCPI command description code examples



1. Example code that you can copy from this table and paste into your own application. Examples are generally shown using the short forms of the commands.
2. Description of the code and what it does. This may also contain the output of the code.

Related commands list

The **Also see** section of the remote command description provides links to commands that are related to the command.

Figure 21: SCPI related commands list example

2260B SCPI command syntax

The 2260B is partially compatible with the IEEE488.2 and SCPI 1999 standard.

Section 4

SCPI command reference

In this section:

Default settings

The following default settings are the factory configuration settings for the power supply.

Function	Default setting
Output	Off
LOCK	0 (Disabled)
Voltage	0 V
Current	0 A
OVP	Maximum
OCP	Maximum

Function	Setting	Default setting
Output ON delay time	F-01	0.00 s
Output OFF delay time	F-02	0.00 s
V-I mode slew rate select	F-03	0 = CV high speed priority
Rising voltage slew rate	F-04	60.00 V/s (2260B-30-XX) 160.0 V/s (2260B-80-XX) 500.0 V/s (2260B-250-XX) 1600 V/s (2260B-800-XX)
Falling voltage slew rate	F-05	60.00 V/s (2260B-30-XX) 160.0 V/s (2260B-80-XX) 500.0 V/s (2260B-250-XX) 1600 V/s (2260B-800-XX)
Rising current slew rate	F-06	72.00 A/s (2260B-30-36) 144.0 A/s (2260B-30-72) 216.0 A/s (2260B-30-108) 27.00 A/s (2260B-80-13) 54.00 A/s (2260B-80-27) 81.00 A/s (2260B-80-40) 9.000 A/s (2260B-250-4) 18.00 A/s (2260B-250-9) 27.00 A/s (2260B-250-13) 2.880 A/s (2260B-800-1) 5.760 A/s (2260B-800-2) 8.640 A/s (2260B-800-4)
Falling current slew rate	F-07	72.00 A/s (2260B-30-36) 144.0 A/s (2260B-30-72) 216.0 A/s (2260B-30-108) 27.00 A/s (2260B-80-13) 54.00 A/s (2260B-80-27) 81.00 A/s (2260B-80-40) 9.000 A/s (2260B-250-4) 18.00 A/s (2260B-250-9) 27.00 A/s (2260B-250-13) 2.880 A/s (2260B-800-1) 5.760 A/s (2260B-800-2) 8.640 A/s (2260B-800-4)
Internal resistance	F-08	0.000 Ω
Bleed resistor circuit control	F-09	1 = ON
Beeper control	F-10	1 = ON
Measurement average	F-17	0 = Low
Lock mode	F-19	0 = Front-panel lock: Allow output off

USB/GPIB settings	Setting	Default setting
Rear panel USB mode	F-22	2 = USB CDC, Auto Detect Speed
GPIB address	F-23	8

LAN settings	Setting	Default setting
LAN	F-36	1 = Enable
DHCP	F-37	1 = Enable
Sockets active	F-57	1 = Enable
Web Server active	F-59	1 = Enable
Web password active	F-60	1 = Enable
Web setting password	F-61	0000

The following settings can only be set during power up.

Power-on configuration	Setting	Default setting
CV Control	F-90	0= Panel control (local)
CC Control	F-91	0= Panel control (local)
Power-on Output	F-92	0 = OFF at startup
Master/Subordinate	F-93	0 = Master/Local
External Out Logic	F-94	0= High ON
Power Switch trip	F-95	0 = Enable

NOTE

Power Switch Trip is not available on all models. Check the power switch for availability.

If the power switch looks like this, the trip function is available: 

If the power switch looks like this, the trip function is not available: 

DISPlay subsystem

This subsystem contains commands that control the front-panel display.

:DISPlay:BLINk

This command causes the display to blink or disables blinking.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:DISPlay:BLINk <n>
:DISPlay:BLINk?
```

<n>	Disable blinking: OFF or 0 Enable blinking: ON or 1

Example

```
DISP:BLIN 1
```

Cause the display to blink.

Also see

None

:DISPlay:MENU[:NAME]

This command selects a screen menu or queries the present screen menu.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:DISPlay:MENU[:NAME] <NR1>
:DISPlay:MENU[:NAME]?
```

<NR1>	The menu: ■ Measurement-Voltage and Measurement-Current: 0 ■ Measurement-Voltage and Measurement-Power: 1 ■ Measurement-Power and Measurement-Current: 2 ■ Set Menu: 3 ■ OVP and OCP Menu: 4 ■ Not used: 5 to 99 ■ F-00 to 99 menu: 100 to 199

Example

```
:DISP:MENU:NAME 0
```

Set the display to the Voltage and Current display screen.

Also see

None

:DISPlay[:WINDow]:TEXT:CLEar

This command clears the text generated by :DISPlay[:WINDow]:TEXT[:DATA] from the main screen.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:DISPlay[:WINDow]:TEXT:CLEar
```

Example

Clear the message from the main screen.

```
:DISP:TEXT:CLE
```

Also see

DISPlay[:WINDow]:TEXT[:DATA] (on page 4-5)

:DISPlay[:WINDow]:TEXT[:DATA]

This command sets or queries text for a display message.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:DISPlay[:WINDow]:TEXT[:DATA] "<string>"
:DISPlay[:WINDow]:TEXT[:DATA]?
```

`<string>`	ASCII characters 20H to 7EH or a null string (`""`)

Details

Writing to the display overwrites the data that is presently on the screen. Overwriting a display area with a shorter string may or may not overwrite the screen.

The string must be enclosed in either double quotes (`" "`) or single quotes (`' '`).

Example

```
:DISP:WIND:TEXT:DATA 'Test 2'
:DISP:WIND:TEXT:DATA?
```

Write `Test 2` to the display.
Return the text data string.

Also see

:DISPlay[:WINDow]:TEXT:CLEar (on page 4-4)

MEASurement subsystem

The commands in the MEASurement subsystem are used to make measurements.

:MEASure[:SCALar]:CURRent[:DC]?

This command makes a measurement and returns the average output current in amperes.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:MEASure[:SCALar]:CURRent[:DC]?
```

Example

```
:MEAS:CURR?
```

Query the average output current.

Also see

None

:MEASure[:SCALar]:POWer[:DC]?

This command makes a measurement and returns the average output power in watts.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:MEASure[:SCALar]:POWer[:DC]?
```

Example

```
:MEAS:POW?
```

Query the average output power.

Also see

None

:MEASure[:SCALar]:VOLTage[:DC]?

This command makes a measurement and returns the average output voltage in volts.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:MEASure[:SCALar]:VOLTage[:DC]?
```

Example

```
:MEAS:VOLT?
```

Query the average output voltage.

Also see

None

OUTPut subsystem

The output subsystem provides information and settings that control the source output.

:APPLy

This command sets the voltage and current.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Voltage: 0 V Current: 0 A

Usage

```
:APPLy <voltage>
:APPLy <voltage>, <current>
:APPLy <MIN|MAX>
:APPLy <MIN|MAX>, <MIN|MAX>
:APPLy?
```

`<voltage>`	Voltage level; the range is 0% to 105% of the rated voltage output
`<current>`	Current level; the range is 0% to 105% of the rated current output

Details

The voltage and current are output when the function is executed.

NOTE

The updated voltage and current values are not reflected on the display until the output is set to On or the `DISPlay:MENU:NAME 3` (set menu) command is sent.

Example

```
APPL 5.05,1.1
APPL?
```

Sets the voltage and current to 5.05 V and 1.1 A.
Query the voltage and current. For this example, the return is:
`+5.050, +1.100`

Also see

DISPlay:MENU[:NAME] (on page 4-4)

:OUTPut:DELay:OFF

This command sets the delay time before output is turned off.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0.00 s

Usage

```
:OUTPut:DELay:OFF <NRf>
:OUTPut:DELay:OFF?
```

<NRf>	Delay time in seconds: 0.00 to 99.99

Details

0 results in no delay.

Example

:OUTP:DEL:OFF 1	Set a delay of 1 second before turning the output off.

Also see

None

:OUTPut:DELay:ON

This command sets the delay time before output is turned on.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0.00 s

Usage

```
:OUTPut:DELay:ON <NRf>
:OUTPut:DELay:ON?
```

<NRf>	Delay time in seconds: 0.00 to 99.99

Details

0 results in no delay.

Example

:OUTP:DEL:ON 1	Set a delay of 1 second before turning the output on.

Also see

None

:OUTPut:MODE

This command selects high speed priority or slew rate priority for CV or CC mode.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 (CV high speed priority)

Usage

```
:OUTPut:MODE <NRf>
:OUTPut:MODE?
```

`<NRf>`	The output mode: ■ CV high speed priority: `CVHS` or `0` ■ CC high speed priority: `CCHS` or `1` ■ CV slew rate priority: `CVLS` or `2` ■ CC slew rate priority: `CCLS` or `3`

Details

This is the equivalent to the F-03 (V-I priority) setting.

Example

`:OUTP:MODE 1`	Selects the CC high speed priority output.

Also see

None

:OUTPut:PROTection:CLEar

This command clears the 2260B protection circuits.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
OUTPut:PROTection:CLEar
```

Details

Clears the overvoltage (OVP), overcurrent (OCP), and overtemperature (OTP) protection circuits. It also clears the shutdown protection circuit.

The AC failure protection cannot be cleared.

Example

`:OUTP:PROT:CLE`	Clears the protection circuits.

Also see

None

:OUTPut:PROTection:TRIPped?

This command returns the state of the protection circuits.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:OUTPut:PROTection:TRIPped?
```

Details

Returns the state of the overvoltage (OVP), overcurrent (OCP), and overtemperature (OTP) protection circuits.

If the protection circuits are not tripped, this query returns `0`.

If the protection circuits tripped, this query returns `1`.

Example

`:OUTP:PROT:TRIP?`	If a protection circuit tripped, the query returns `1`. If the protection circuit is not tripped, the query returns `0`.

Also see

None

:OUTPut[:STATe]:TRIGgered

This command turns the output on or off when a software trigger is generated.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:OUTPut[:STATe]:TRIGgered <state>
:OUTPut[:STATe]:TRIGgered?
```

`<state>`	Turn source off when a software trigger is generated: `OFF` or `0` Turn source on when a software trigger is generated: `ON` or `1`

Example

`:OUTP:TRIG ON`	Switch the source output of the instrument to ON when a software trigger is generated.

Also see

None

:OUTPut[:STATe][:IMMediate]

This command enables or disables the source output.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 (OFF)

Usage

```
:OUTPut[:STATe][:IMMediate] <state>
:OUTPut[:STATe][:IMMediate]?
```

`<state>`	Turn source off: `OFF` or `0` Turn source on: `ON` or `1`

Example

`:OUTP ON`	Switch the source output of the instrument to `ON`.

Also see

None

SENSe subsystem

The command in the SENSe subsystem determines the level of smoothing for the average setting.

:SENSe:AVERage:COUNt

This command determines the level of smoothing for the average setting.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 (LOW)

Usage

```
:SENSe:AVERage:COUNt <value>
:SENSe:AVERage:COUNt?
```

`<value>`	The level of smoothing: ■ Low level: `LOW` or `0` ■ Middle level: `MIDDle` or `1` ■ High level: `HIGH` or `2`

Details

This is equivalent to the F-17 function setting.

Smoothing affects the measurement time. For example, if the output voltage changes, the time for the measurements to update for the three settings is approximately:

- Low: 0.25 s
- Middle: 1.5 s
- High: 2.5 s

Only available for firmware version 1.5 or above. You can use :SYSTem:INFormation? (on page 4-38) to check your firmware version number.

Example

`SENS:AVER:COUN MIDD`	Sets the level of smoothing to middle.

Also see

None

SOURce subsystem

The commands in the SOURce subsystem configure and control the current source and voltage source.

[:SOURce]:CURRent:PROTection:STATe

This command sets overcurrent protection (OCP) on or off.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
[:SOURce]:CURRent:PROTection:STATe <state>
[:SOURce]:CURRent:PROTection:STATe?
```

`<state>`	Turn OCP off: `OFF` or `0` Turn OCP on: `ON` or `1`

Details

The query returns the state of OCP.

Example

`:CURR:PROT:STAT OFF`	Set OCP off.

Also see

None

[:SOURce]:CURRent:PROTection[:LEVel]

This command sets or queries the overcurrent protection (OCP) setting in amperes.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	MAX

Usage

```
[:SOURce]:CURRent:PROTection[:LEVel] <value>
[:SOURce]:CURRent:PROTection[:LEVel] <MIN|MAX>
[:SOURce]:CURRent:PROTection[:LEVel]?
[:SOURce]:CURRent:PROTection[:LEVel]? <MIN|MAX>
```

`<value>`	The OCP level; the range is 10% to 110% of the rated current output level

Example

`:CURR:PROT? MIN`	Returns the minimum possible current level in amperes. Example output: `+3.600`

Also see

None

[:SOURce]:CURRent:SLEW:FALLing

This command specifies and queries the falling slew rate for the current source.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	2260B-30-36: 72.00A/s 2260B-30-72: 144.0A/s 2260B-30-108: 216.0A/s 2260B-80-13: 27.00A/s 2260B-80-27: 54.00A/s 2260B-80-40: 81.00A/s 2260B-250-4: 9.000A/s 2260B-250-9: 18.00A/s 2260B-250-13: 27.00A/s 2260B-800-1: 2.880A/s 2260B-800-2: 5.760A/s 2260B-800-4: 8.640A/s

Usage

```
[:SOURce]:CURRent:SLEW:FALLing <value>
[:SOURce]:CURRent:SLEW:FALLing <MIN|MAX>
[:SOURce]:CURRent:SLEW:FALLing?
[:SOURce]:CURRent:SLEW:FALLing? <MIN|MAX>
```

`<value>`	The current falling slew rate; see **Details**

Details

This is only applicable for CC slew rate priority mode.

Slew rate (A per second)	Model
0.01 to 72.00	2260B-30-36
0.1 to 144.0	2260B-30-72
0.1 to 216.0	2260B-30-108
0.01 to 27.00	2260B-80-13 2260B-250-13
0.01 to 54.00	2260B-80-27
0.01 to 81.00	2260B-80-40
0.001 to 9.000	2260B-250-4
0.01 to 18.00	2260B-250-9
0.001 to 2.880	2260B-800-1
0.001 to 5.760	2260B-800-2
0.001 to 8.640	2260B-800-4

Example

`:CURR:SLEW:FALL 1`	Sets the falling current slew rate to 1 A per second.

Also see

None

[:SOURce]:CURRent:SLEW:RISing

This command specifies and queries the rising slew rate for the current source.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	2260B-30-36: 72.00A/s 2260B-30-72: 144.0A/s 2260B-30-108: 216.0A/s 2260B-80-13: 27.00A/s 2260B-80-27: 54.00A/s 2260B-80-40: 81.00A/s 2260B-250-4: 9.000A/s 2260B-250-9: 18.00A/s 2260B-250-13: 27.00A/s 2260B-800-1: 2.880A/s 2260B-800-2: 5.760A/s 2260B-800-4: 8.640A/s

Usage

```
[:SOURce]:CURRent:SLEW:RISing <value>
[:SOURce]:CURRent:SLEW:RISing <MIN|MAX>
[:SOURce]:CURRent:SLEW:RISing?
[:SOURce]:CURRent:SLEW:RISing? <MIN|MAX>
```

`<value>`	The current rising slew rate; see **Details**

Details

This is only applicable for CC slew rate priority mode.

Slew rate (A per second)	Model
0.01 to 72.00	2260B-30-36
0.1 to 144.0	2260B-30-72
0.1 to 216.0	2260B-30-108
0.01 to 27.00	2260B-80-13 2260B-250-13
0.01 to 54.00	2260B-80-27
0.01 to 81.00	2260B-80-40
0.001 to 9.000	2260B-250-4
0.01 to 18.00	2260B-250-9
0.001 to 2.880	2260B-800-1
0.001 to 5.760	2260B-800-2
0.001 to 8.640	2260B-800-4

Example

`:CURR:SLEW:RIS 72`	Sets the rising current slew rate to 72 A per second.

Also see

None

[:SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude]

This command sets or queries the current level in amperes when a software trigger has been generated.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
[:SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>
[:SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude] <MIN|MAX>
[:SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude]?
[:SOURce]:CURRent[:LEVel]:TRIGgered[:AMPLitude]? <MIN|MAX>
```

`<value>`	Current level; the range is 0% to 105% of the rated current output

Example

`:CURR:TRIG? MAX`	Returns the maximum possible current level in amps; an example return is: `37.800`

Also see

None

[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]

This command sets or queries the current level in amperes.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 A

Usage

```
[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude] <value>
[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude] <MIN|MAX>
[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]?
[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]? <MIN|MAX>
```

`<value>`	Current level; the range is 0% to 105% of the rated current output

Details

For externally set current levels (from the analog control connector), the set current level is returned.

Example

`:CURR? MAX`	Returns the maximum possible current level in amps, such as: `37.800`

Also see

None

[:SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]

This command sets or queries the internal resistance in ohms.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0.000 Ω

Usage

```
[:SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>
[:SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude] <MIN|MAX>
[:SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]?
[:SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]? <MIN|MAX>
```

`<value>`	The internal resistance; see **Details**

Details

When the internal resistance is set, it is a resistance in series with the positive output terminal. This allows the power supply to simulate power sources that have internal resistances, such as lead acid batteries. The internal resistance range for each 2260B model is shown in the following table.

Internal resistance range

Range in ohms	Model
0.000 to 0.833	2260B-30-36
0.000 to 0.417	2260B-30-72
0.000 to 0.278	2260B-30-108
0.000 to 5.926	2260B-80-13
0.000 to 2.963	2260B-80-27
0.000 to 1.975	2260B-80-40
0.00 to 55.55	2260B-250-4
0.00 to 27.77	2260B-250-9
0.0 to 18.51	2260B-250-13
0.0 to 555.5	2260B-800-1
0.0 to 277.8	2260B-800-2
0.0 to 185.1	2260B-800-4

Example

`:RES 0.1`	Sets the internal resistance to 100 mΩ.

Also see

None

[:SOURce]:VOLTage:PROTection[:LEVel]

This command sets or queries the overvoltage protection (OVP) level.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	MAX

Usage

```
[:SOURce]:VOLTage:PROTection[:LEVel] <value>
[:SOURce]:VOLTage:PROTection[:LEVel] <MIN|MAX>
[:SOURce]:VOLTage:PROTection[:LEVel]?
[:SOURce]:VOLTage:PROTection[:LEVel]? <MIN|MAX>
```

`<value>`	Voltage level; 0% to 110% of the rated output voltage level

Example

`:VOLT:PROT MAX`	Sets the OVP level to its maximum.

Also see

None

[:SOURce]:VOLTage:SLEW:FALLing

This command specifies and queries the falling slew rate for the voltage source.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	2260B-30-xx: 60.00 V/s 2260B-80-xx: 160.0 V/s 2260B-250-xx: 500.0 V/s 2260B-800-xx: 1600 V/s

Usage

```
[:SOURce]:VOLTage:SLEW:FALLing <value>
[:SOURce]:VOLTage:SLEW:FALLing <MIN|MAX>
[:SOURce]:VOLTage:SLEW:FALLing?
[:SOURce]:VOLTage:SLEW:FALLing? <MIN|MAX>
```

`<value>`	The voltage falling slew rate; see **Details**

Details

This is only applicable for CV slew rate priority mode.

Slew rate (volts per second)	Model
0.01 to 60.00	2260B-30-*xx*
0.1 to 160.0	2260B-80-*xx*
0.1 to 500.0	2260B-250-*xx*
1 to 1600	2260B-800-*xx*

Example

`:VOLT:SLEW:FALL 1`	Sets the falling voltage slew rate to 1 V/s.

Also see

None

[:SOURce]:VOLTage:SLEW:RISing

This command specifies and queries the rising slew rate for the voltage source.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	2260B-30-xx: 60.00 V/s 2260B-80-xx: 160.0 V/s 2260B-250-xx: 500.0 V/s 2260B-800-xx: 1600 V/s

Usage

```
[:SOURce]:VOLTage:SLEW:RISing <value>
[:SOURce]:VOLTage:SLEW:RISing <MIN|MAX>
[:SOURce]:VOLTage:SLEW:RISing?
[:SOURce]:VOLTage:SLEW:RISing? <MIN|MAX>
```

`<value>`	The voltage rising slew rate; see **Details**

Details

This is only applicable for CV slew rate priority mode.

Slew rate (volts per second)	Model
0.01 to 60.00	2260B-30-*xx*
0.1 to 160.0	2260B-80-*xx*
0.1 to 500.0	2260B-250-*xx*
1 to 1600	2260B-800-*xx*

Example

`:VOLT:SLEW:RIS MAX`	Sets the rising voltage slew rate to its maximum.

Also see

None

[:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude]

This command sets or queries the voltage level in volts when a software trigger has been generated.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
[:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>
[:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude] <MIN|MAX>
[:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
[:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude]? <MIN|MAX
```

`<value>`	Voltage level; 0% to 105% of the rated output voltage level

Example

`:VOLT:TRIG 10`	Sets the voltage level to 10 volts when a software trigger is generated.

Also see

None

[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]

This command sets or queries the voltage level in volts.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 V

Usage

```
[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <value>
[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <MIN|MAX>
[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?
[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]? <MIN|MAX>
```

`<value>`	Voltage level; 0% to 105% of the rated output voltage level

Example

`VOLT 10`	Sets the voltage level to 10 volts.

Also see

None

STATus subsystem

The STATus subsystem controls the status registers of the 2260B. For additional information on the status model, see Status model (on page 6-1).

:STATus:OPERation:CONDition?

This command reads the Operation Status register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:OPERation:CONDition?
```

Details

Returns the bit sum of the Operation Condition register. This query does not clear the register.

Example

`:STAT:OPER:COND?`	Returns the bit sum of the Operation Condition register.

Also see

None

:STATus:OPERation:ENABle

This command sets or reads the bit sum of the Operation Status Enable register of the status model.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x0000

Usage

```
:STATus:OPERation:ENABle <n>
:STATus:OPERation:ENABle?
```

<n>	The status of the Operation Status Enable register: 0 to 32767

Also see

None

:STATus:OPERation:NTRansition

This command sets or queries the bit sum of the negative transition filter of the Operation Status register.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x0000

Usage

```
:STATus:OPERation:NTRansition <NRf>
:STATus:OPERation:NTRansition?
```

<NRf>	The bit sum: 0 to 32767

Also see

None

:STATus:OPERation:PTRansition

This command sets or queries the bit sum of the positive transition filter of the Operation Status register.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x7FFF (32767)

Usage

```
:STATus:OPERation:PTRansition <NRf>
:STATus:OPERation:PTRansition?
```

<NRf>	The bit sum: 0 to 32767

Also see

None

:STATus:OPERation[:EVENt]?

This command reads and clears the Operation Event Register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:OPERation[:EVENt]?
```

Details

This command reads the Operation Event Register of the status model and then clears the register.

The instrument returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register.

Also see

None

:STATus:PRESet

This command returns registers to their default conditions.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:PRESet
```

Details

This command resets the Enable register and the PTRansition and NTRansition filters in the Operation Status and the Questionable Status registers. The registers and filters are reset to a default value.

Register or filter	Default value
QUEStionable Status Enable	0x0000
QUEStionable Status Positive Transition	0x7FFF
QUEStionable Status Negative Transition	0x0000
Operation Status Enable	0x0000
Operation Status Positive Transition	0x7FFF
Operation Status Negative Transition	0x0000

The Questionable Status Enable registers and the Operation Status Enable registers are reset to `0` (only positive transitions are recognized). The Questionable Status and Operation Status Positive Transition filters are all set high (`0x7FFF`) and the Negative Transition filters are all set low (`0x0000`).

Also see

None

:STATus:QUEStionable:CONDition?

This command reads the Questionable Status register of the status model.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEStionable:CONDition?
```

Details

This query does not clear the register.

The return is `0` to `32767`.

Also see

None

:STATus:QUEStionable:ENABle

This command sets or reads the bit sum of the Questionable Status Enable register of the status model.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x0000

Usage

```
:STATus:QUEStionable:ENABle <n>
:STATus:QUEStionable:ENABle?
```

`<n>`	The status of the Questionable Status Enable register : `0` to `32767`

Also see

None

:STATus:QUEStionable:NTRansition

This command sets or queries the bit sum of the negative transition filter of the Questionable Status register.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x0000

Usage

```
:STATus:QUEStionable:NTRansition <NRf>
:STATus:QUEStionable:NTRansition?
```

<NRf>	The bit sum: 0 to 32767

Also see

None

:STATus:QUEStionable:PTRansition

This command sets or queries the bit sum of the positive transition filter of the Questionable Status register.

Type	Affected by	Where saved	Default value
Command and query	:STATus:PRESet	Not applicable	0x7FFF (32767)

Usage

```
:STATus:QUEStionable:PTRansition <NRf>
:STATus:QUEStionable:PTRansition?
```

<NRf>	The bit sum: 0 to 32767

Also see

None

:STATus:QUEStionable[:EVENt]?

This command reads the Questionable Status Event register.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:STATus:QUEStionable[:EVENt]?
```

Details

This command reads the register and then clears the contents of the register.

Returns the bit sum from 0 to 32767.

Example

`:STAT:QUES?`	Returns the contents of Questionable Status Event register.

Also see

None

SYSTem subsystem

This subsystem contains commands that affect the overall operation of the instrument.

:SYSTem:COMMunicate:ENABle

This command enables or disables LAN, GPIB, or USB remote interfaces and remote services such as Sockets and Web Server.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Nonvolatile memory	0

Usage

```
:SYSTem:COMMunicate:ENABle <mode>, <interface>
:SYSTem:COMMunicate:ENABle? <interface>
```

`<mode>`	Turn the selected interface on: `ON` or `1`; see **Details** Turn the selected interface off: `OFF` or `0` Select full speed (USB only): `FULL` or `2`
`<interface>`	The interface: ■ GPIB: `GPIB` ■ USB: `USB` ■ LAN: `LAN` ■ Sockets (firmware v1.12 and later): `SOCKets` ■ Web server: `WEB`

Details

This setting is applied after the power is cycled.

When the interface is set to `USB` and the mode is set to `ON`, the USB speed is autodetected.

Example 1

`:SYST:COMM:ENAB 1, USB` `:SYST:COMM:ENAB? USB`	Turns the USB interface on. USB speed is autodetected. The return from the query is `1` (USB is on and autodetected).

Example 2

`:SYST:COMM:ENAB 2, USB`	Turns the USB interface on. USB speed is full speed only.

Also see

None

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

This command sets or queries the GPIB address.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	8

Usage

```
:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <NR1>
:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?
```

<NR1>	The GPIB address: 0 to 30

Details

This setting is applied after the power is cycled.

Example

SYST:COMM:GPIB:ADDR 15	Set the GPIB address to 15.

Also see

None

:SYSTem:COMMunicate:LAN:DHCP

This command turns DHCP on or off.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	1 (ON)

Usage

```
:SYSTem:COMMunicate:LAN:DHCP <state>
:SYSTem:COMMunicate:LAN:DHCP?
```

<state>	Turn DHCP off: OFF or 0 Turn DHCP on: ON or 1

Details

This setting is applied after the power is cycled.

Also see

None

:SYSTem:COMMunicate:LAN:GATEway

This command sets or queries the LAN gateway address for the instrument.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Nonvolatile memory	"0.0.0.0"

Usage

```
:SYSTem:COMMunicate:LAN:GATEway "<string>"
:SYSTem:COMMunicate:LAN:GATEway?
```

`<string>`	Gateway address in string format (`"address"`); `0` to `255` for each part of the address; applicable ASCII characters: `20H` to `7EH`

Details

This setting is applied after the power is cycled.

Example

```
:SYST:COMM:LAN:GATE "172.16.0.254"
```

Sets the LAN gateway to `172.16.0.254`.

Also see

None

:SYSTem:COMMunicate:LAN:IPADdress

This command sets or queries the LAN IP address.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Nonvolatile memory	0.0.0.0

Usage

```
:SYSTem:COMMunicate:LAN:IPADdress "<string>"
:SYSTem:COMMunicate:LAN:IPADdress?
```

`<string>`	Set the LAN IP address in string format; `0` to `255` for each part of the address; applicable ASCII characters are `20H` to `7EH`

Details

This setting is applied after the power is cycled.

Example

`SYST:COMM:LAN:IPAD "172.16.5.111"`	Set the LAN IP address to `172.16.5.111`.

Also see

None

:SYSTem:COMMunicate:LAN:SMASk

This command sets or queries the LAN subnet mask for the instrument.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Nonvolatile memory	AUTO

Usage

```
:SYSTem:COMMunicate:LAN:SMASk "<string>"
:SYSTem:COMMunicate:LAN:SMASk?
```

`<string>`	Subnet mask in string format ("`mask`"); `0` to `255` for each part of the address; applicable ASCII characters: `20H` to `7EH`

Details

This setting is applied after the power is cycled.

Example

```
:SYST:COMM:LAN:SMAS "255.255.0.0"
```

Sets the LAN mask to 255.255.0.0

Also see

None

:SYSTem:COMMunicate:LAN:WEB:PACTive

This command sets or queries whether the web password is on or off.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	1 (ON)

Usage

```
:SYSTem:COMMunicate:LAN:WEB:PACTive <state>
:SYSTem:COMMunicate:LAN:WEB:PACTive?
```

`<state>`	Web password off: `OFF` or `0` Web password on: `ON` or `1`

Details

This setting is applied after the power is cycled.

Example

`:SYST:COMM:LAN:WEB:PACT 0`	Set the web password off.

Also see

None

:SYSTem:COMMunicate:LAN:WEB:PASSword

This command sets or queries the web password.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0000

Usage

```
:SYSTem:COMMunicate:LAN:WEB:PASSword <NR1>
:SYSTem:COMMunicate:LAN:WEB:PASSword?
```

<NR1>	The web password: 0000 to 9999

Details

This setting is applied after the power is cycled.

Example

:SYST:COMM:LAN:WEB:PASS 1234	Set the web password to 1234.

Also see

None

:SYSTem:COMMunicate:LAN:DNS

This command sets or queries the DNS address.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:COMMunicate:LAN:DNS "<string>"
:SYSTem:COMMunicate:LAN:DNS?
```

<string>	DNS in string format ("mask"); 0 to 255 for each part of DNS; applicable ASCII characters: 20H to 7EH

Details

This setting is applied after the power is cycled.

Example

```
SYST:COMM:LAN:DNS "172.16.1.252"
```

Sets the DNS to 172.16.1.252.

Also see

None

:SYSTem:COMMunicate:LAN:HOSTname?

This command queries the LAN host name.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:COMMunicate:LAN:HOSTname?
```

Details

The return is the host name in string format.

Example

`:SYST:COMM:LAN:HOST?`	Returns the host name. For example, you might see: `P-160054`

Also see

None

:SYSTem:COMMunicate:LAN:MAC?

This command queries the LAN MAC address.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:COMMunicate:LAN:MAC?
```

Details

The MAC address is a character string representing the MAC address of the instrument in hexadecimal notation. The string is returned in the format `"FF-FF-FF-FF-FF-FF"`.

The MAC address cannot be changed.

Example

`:SYST:COMM:LAN:MAC?`	Returns the MAC address. For example, you might see: `02-80-AD-20-31-B1`

Also see

None

:SYSTem:COMMunicate:USB:FRONt:STATe?

This command queries the front-panel USB-A port state.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:COMMunicate:FRONt:STATe?
```

Details

The response to the query indicates if a USB drive is inserted into the USB port:

- Absent: `0`
- Present: `1`

Example

`:SYST:COMM:USB:FRON:STAT?`	If a USB flash drive is inserted in the port, the return is `1`.

Also see

None

:SYSTem:COMMunicate:USB:REAR:STATe?

This command queries the rear-panel USB-B port state.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:COMMunicate:REAR:STATe?
```

Details

The response to the query indicates the connections to the USB port:

- Disabled: `0`
- GPIB-USB adapter: `1`
- USB CDC: `2`

Example

`:SYST:COMM:USB:REAR:STAT?`	Example if a GPIB adapter is attached: `1`

Also see

:SYSTem:COMMunicate:ENABle (on page 4-25)

:SYSTem:CONFigure:BEEPer[:STATe]

This command turns the beeper on or off.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	1 (ON)

Usage

```
:SYSTem:CONFigure:BEEPer[:STATe] <state>
:SYSTem:CONFigure:BEEPer[:STATe]?
```

`<state>`	The status of the beeper: ■ Disable the beeper: `OFF` or `0` ■ Enable the beeper: `ON` or `1`

Example

`:SYST:CONF:BEEP 0`	Set the beeper to off.

Also see

None

:SYSTem:CONFigure:BLEeder[:STATe]

This command enables or disables the bleed resistor.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	1 (ON)

Usage

```
:SYSTem:CONFigure:BLEeder[:STATe] <state>
:SYSTem:CONFigure:BLEeder[:STATe]?
```

`<state>`	The status of the bleed resistor: ■ Disable the resistor: `OFF` or `0` ■ Enable the resistor: `ON` or `1` ■ Turn the bleed resistor on or off automatically: `AUTO` or `2`

Details

When `AUTO` is selected, the bleed resistor is automatically turned on when the output is turned on and turned off when the output is turned off. The `AUTO` setting is only applicable to firmware version 1.59 or above. You can use :SYSTem:INFormation? (on page 4-38) to check the firmware version number.

Example

`:SYST:CONF:BLE AUTO`	Set the bleed resistor to automatically turn on and off.

Also see

None

:SYSTem:CONFigure:BTRip:PROTection

This command enables or disables power off when the overvoltage protection (OVP), overcurrent protection (OCP), or overtemperature protection (OTP) setting is tripped.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (Enable)

Usage

```
:SYSTem:CONFigure:BTRip:PROTection <state>
:SYSTem:CONFigure:BTRip:PROTection?
```

`<state>`	Disable the power switch trip for OVP, OCP, or OTP: `DISable` or `1` Enable the power switch trip for OVP, OCP, of OTP: `ENABle` or `0`

Details

This setting only applies after power has been cycled.

NOTE

Power Switch Trip is not available on all models. Check the power switch for availability.

If the power switch looks like this, the trip function is available: 

If the power switch looks like this, the trip function is not available: 

If the trip function is not available, this command does nothing. When OVP, OCP, or OTP is detected, the output automatically turns off regardless of the setting of this command.

Example

`:SYST:CONF:BTR:PROT DIS`	Switch the power off when a protection setting is tripped.

Also see

None

:SYSTem:CONFigure:BTRip[:IMMediate]

This command turns the power switch off immediately.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:CONFigure:BTRip[:IMMediate]
```

Details

The ability to turn the power switch off immediately is not available on all models. Check the power switch for availability.

If the power switch looks like this, the BTRip command is available: 

If the power switch looks like this, the BTRip command is not available:  . If the command is sent, it is ignored.

Also see

None

:SYSTem:CONFigure:CURRent:CONTrol

This command sets or queries the CC control mode (local control, external voltage control, external resistance control).

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (local control)

Usage

```
:SYSTem:CONFigure:CURRent:CONTrol <setting>
:SYSTem:CONFigure:CURRent:CONTrol?
```

`<setting>`	The CC control mode: ■ Local (panel) control: `0` ■ External voltage control: `1` ■ External resistance increasing control (10 kΩ is I_o maximum; 0 kΩ is the I_o minimum): `2` ■ External resistance decreasing control (10 kΩ is I_o minimum; 0 kΩ is I_o maximum): `3`

Details

This setting is applied only after power has been cycled.

Example

`:SYST:CONF:CURR:CONT 1`	Set CC control mode to external voltage control.

Also see

None

:SYSTem:CONFigure:MSLave

This command sets or queries the instrument operation mode.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (Master/Local)

Usage

```
:SYSTem:CONFigure:MSLave <setting>
:SYSTem:CONFigure:MSLave?
```

`<setting>`	The operation mode: ■ Master/Local control: `0` ■ Master/Parallel 1 (2 units): `1` ■ Master/Parallel 2 (3 units): `2` ■ Subordinate/Parallel: `3` ■ Subordinate/Series (only 30 V and 80 V models): `4`

Details

This setting is applied only after power has been cycled.

Example

`:SYST:CONF:MSL 1`	Use the Master/Parallel 1 operation mode with two instruments.

Also see

None

:SYSTem:CONFigure:OUTPut:EXTernal[:MODE]

This command sets the external logic to active high or active low.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (HIGH)

Usage

```
:SYSTem:CONFigure:OUTPut:EXTernal[:MODE] <state>
:SYSTem:CONFigure:OUTPut:EXTernal[:MODE]?
```

`<state>`	The setting for the external logic: ■ Active high: `HIGH` or `0` ■ Active low: `LOW` or `1`

Details

This setting is applied after the power is cycled.

Example

`:SYST:CONF:OUTP:EXT 0`	Set the external logic to active high.

Also see

None

:SYSTem:CONFigure:OUTPut:PON[:STATe]

This command turns the output on or off at power-up.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (OFF)

Usage

```
:SYSTem:CONFigure:OUTPut:PON[:STATe] <state>
:SYSTem:CONFigure:OUTPut:PON[:STATe]?
```

`<state>`	■ Output off at power-up: `OFF` or `0` ■ Output on at power-up: `ON` or `1`

Details

This setting is applied after the power is cycled.

Example

`:SYST:CONF:OUTP:PON 0`	Set the output off at power-up.

Also see

None

:SYSTem:CONFigure:VOLTage:CONTrol

This command sets or queries the CV control mode (local control, external voltage control, external resistance control).

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Nonvolatile memory	0 (local panel control)

Usage

```
:SYSTem:CONFigure:VOLTage:CONTrol <setting>
:SYSTem:CONFigure:VOLTage:CONTrol?
```

`<setting>`	The CV control mode: ■ Local (panel) control: `0` ■ External voltage control: `1` ■ External resistance control increasing (10 kΩ is V_o maximum; 0 kΩ is the V_o minimum): `2` ■ External resistance control decreasing (10 kΩ is V_o minimum; 0 kΩ is V_o maximum): `3`

Details

This setting is applied only after power has been cycled.

Example

`:SYST:CONF:CURR:CONT 1`	Set CV control mode to use external voltage control.

Also see

None

:SYSTem:ERRor?

This command queries the error queue.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:ERRor?
```

Details

The last error message is returned and cleared. A maximum of 16 errors are stored in the error queue. Each remote interface I/O session (GPIB, USB, and LAN) has its own error queue.

The return is an error code followed by an error message as a string.

Example

`:SYST:ERR?`	Example return: `-100, "Command error"`

Also see

None

:SYSTem:INFormation?

This command returns system information.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:INFormation?
```

Details

This command returns the machine version, build date, keyboard CPLD version and analog CPLD version. It is returned as definite length arbitrary block response data in the format:

```
#3238MFRS XXXXXX,Model AAAAAA,SN BBBBBB,Firmware-Version CCCCCCCCC,Keyboard- CPLD
   FFFFFF,AnalogControl-CPLD GGGGGG,Kernel- BuilTON YYYY-MM-DD,MAC HH-HH-HH-HH-HH-
   HH
```

Where:

- *XXXXX*: Manufacturer
- *AAAAAA*: Model number
- *BBBBBB*: Serial number
- *CCCCCCCCC*: Firmware version
- *FFFFFF*: Keyboard CPLD version
- *GGGGGG*: Analog board CPLD version
- *MM DD YYYY*: Month, date, and year of the firmware build
- *HH-HH-HH-HH-HH-HH*: MAC address

Example

```
:SYST:INF?
```

An example return is:

```
#3183MFRS Keithley Instruments Inc., Model 2260B-30-72,SN 123456789,Firmware-
   Version 01.89.20220809,Keyboard-CPLD 0x30c,AnalogControl-CPLD 0x427,Kernel-
   BuiltON 2022-1-13,MAC 00-00-0a-00-00-00
```

Also see

None

:SYSTem:KEYLock:MODE

This command sets the behavior of the Output key when the key panel lock is on.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 (Allow output to be turned off)

Usage

```
:SYSTem:KEYLock:MODE <mode>
:SYSTem:KEYLock:MODE?
```

`<mode>`	The key lock mode when panel lock is on: ■ Allow the output to be turned off: `0` ■ Allow output to be turned on or off: `1`

Details

This setting is the equivalent of the F-19 function setting.

Example

`:SYST:KEYL:MODE 0`	Allow the output to be turned off when the panel is locked.

Also see

:SYSTem:KLOCk (on page 4-39)

:SYSTem:KLOCk

This command enables or disables the front-panel key lock.

Type	Affected by	Where saved	Default value
Command and query	*RST :SYSTem:PRESet Factory set value (F-88)	Not applicable	0 (OFF)

Usage

```
:SYSTem:KLOCk <state>
:SYSTem:KLOCk?
```

`<state>`	The key lock state: ■ Unlock front-panel keys: `0` or `OFF` ■ Lock front-panel keys: `1` or `ON`

Example

`:SYST:KLOC 0`	Unlock the front-panel keys.

Also see

:SYSTem:KEYLock:MODE (on page 4-39)

:SYSTem:PRESet

This command resets instrument settings to their factory defaults.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:PRESet
```

Details

This command is identical in effect to the `*RST` command.

For a list of the default settings, refer to Default settings (on page 4-1).

Also see

*RST (on page 5-4)

:SYSTem:VERSion?

This command returns the version of the SCPI specifications that the instrument complies with.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
:SYSTem:VERSion?
```

Example

`:SYST:VERS?`	Example return: `1999.0`

Also see

None

TRIGger subsystem

The trigger commands generate and configure software triggers. This power supply supports the following trigger functions:

- **Transient:** Specifies the current and voltage settings in advance and uses the trigger to set them. Refer to [:SOURce]:CURRent[:LEVel]:TRIGgered [:AMPLitude] (on page 4-15) and [:SOURce]:VOLTage[:LEVel]:TRIGgered[:AMPLitude] (on page 4-19).
- **Output:** Specifies the output on and off settings in advance and uses the trigger to set them. Refer to OUTPut[:STATe]:TRIGgered (on page 4-10).

:ABORt

This command cancels any triggered actions.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:ABORt
```

Also see

None

:INITiate[:IMMediate]:NAME

This command starts the transient or output trigger.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:INITiate[:IMMediate]:NAME <n>
```

<n>	Start the transient trigger: `TRANsient` Start the output trigger: `OUTPut`

Example

Start the transient trigger.

```
:INIT:NAME TRAN
```

Also see

None

:TRIGger:TRANsient[:IMMediate]

This command generates a software trigger for the transient trigger system.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
TRIGger:TRANsient[:IMMediate]
```

Example

Generate the software trigger for the transient trigger system.

```
:TRIG:TRAN
```

Also see

None

:TRIGger:TRANsient:SOURce

This command sets or queries the trigger source for the transient system.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	IMMediate

Usage

```
:TRIGger:TRANsient:SOURce <value>
:TRIGger:TRANsient:SOURce?
```

<value>	Internal software trigger: BUS Start the trigger immediately: IMMediate

Details

When BUS is selected, the instrument waits for the *TRG or the IEEE 488.1 GET group execute trigger command to start the trigger.

Example: Immediate

```
:TRIG:TRAN:SOUR IMM
:CURR:TRIG MAX
:VOLT:TRIG 5
:INIT:NAME TRAN
```

Set the trigger source for the transient system to immediate.
Set the current to the maximum and the voltage to 5 V.
Initiate a trigger.

Example: Bus

```
:TRIG:TRAN:SOUR BUS
:CURR:TRIG MAX
:VOLT:TRIG 5
:INIT:NAME TRAN
:TRIG:TRAN
```

Set the trigger source for the transient system to bus.
Set the current to the maximum and the voltage to 5 V.
Initiate a trigger.

Also see

None

:TRIGger:OUTPut[:IMMediate]

This command generates a software trigger for the output trigger system.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
:TRIGger:OUTPut[:IMMediate]
```

Also see

None

:TRIGger:OUTPut:SOURce

This command sets or queries the trigger source for the output system.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
:TRIGger:OUTPut:SOURce <NRf>
:TRIGger:OUTPut:SOURce?
```

<NRf>	Internal software trigger: BUS Start the trigger immediately: IMMediate

Details

When BUS is selected, the instrument waits for the *TRG or the IEEE 488.1 GET group execute trigger command to start the trigger.

Example: Immediate

```
:TRIG:OUTP:SOUR IMM
:OUTP:TRIG 1
:INIT:NAME OUTP
```

Turn the output on.

Example: Bus

```
:TRIG:OUTP:SOUR BUS
:OUTP:TRIG 1
:INIT:NAME OUTP
:TRIG:OUTP
```

Turn the output on.

Also see

None

Section 5

Common commands

In this section:

Introduction

The common commands are device commands that are common to all devices on the bus. These commands are designated and defined by the IEEE-488.2 standard.

*CLS

This command clears the event registers and queues.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*CLS
```

Details

Use the `*CLS` command to clear (reset to `0`) the bits of the following registers in the 2260B:

- Standard Event Status
- Operation Status
- Questionable Status

The corresponding Enable registers in each of the above registers are not cleared.

If a `<NL>` newline code immediately precedes a `*CLS` command, the Error Queue and the MAV bit in the Status Byte Register is also cleared.

Example

```
*CLS
```

Clear the event registers and queues.

Also see

Status model (on page 6-1)

*ESE

This command sets and queries bits in the Standard Event Status Enable Register.

Type	Affected by	Where saved	Default value
Command and query	Power cycle	Not saved	See **Details**

Usage

```
*ESE <mask>
*ESE?
```

`<mask>`	Value in the range from `0` through `255`

Details

The binary bits of the Standard Event Enable Register are set according to this value.

The query returns the bit sum of the Standard Event Status Enable register.

Example

```
*ESE 145
*ESE?
```

Sets the Standard Event Enable Register to binary `10010001`, which enables the PON, EXE, and OPC bits. An example of the return to the query is 186, which shows that the Standard Event Enable Register contains the binary value `10111010`.

Also see

None

*ESR?

This command reads and clears the contents of the Standard Event Status Register.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*ESR?
```

Details

The instrument returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register and clears the Standard Event Status Register.

Example

```
*ESR?
```

Query the contents of the Standard Event Status Register (SESR).
Example output:
`145`
This output indicates that the SESR contains the binary value `10010001`.

Also see

None

*IDN?

This command retrieves the identification string of the instrument.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*IDN?
```

Details

The identification string includes the manufacturer, model number, serial number, and firmware revision of the instrument. The string is formatted as follows:

```
KEITHLEY INSTRUMENTS,MODEL nnnn,xxxxxxxx,yyyyyy
```

Where:

- *nnnn* is the model number
- *xxxxxxxx* is the serial number
- *yyyyyy* is the firmware revision

Example

```
*IDN?
```

Example output:
```
KEITHLEY INSTRUMENTS INC.,MODEL 2260B-30-72, TW123456, 01.00.20110101
```

Also see

None

*OPC

This command sets the operation complete (OPC) bit after all pending commands, including overlapped commands, have been executed.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
*OPC
*OPC?
```

Details

When `*OPC` is sent, the OPC bit (bit 0) in the Status Event Status Register is set after all pending command operations have been executed.

`*OPC?` returns `1` when all outstanding commands have completed.

Example

```
*OPC?
```

Returns `1` to indicate that all pending OPC operations are finished.

Also see

None

*RST

This command resets the instrument settings to their default values.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*RST
```

Details

This command is identical in effect to the `:SYSTem:PRESet` command.

For a list of the default settings, refer to Default settings (on page 4-1).

Also see

:SYSTem:PRESet (on page 4-40)

*SRE

This command sets or clears the bits of the Service Request Enable register.

Type	Affected by	Where saved	Default value
Command and query	Not applicable	Not applicable	Not applicable

Usage

```
*SRE <NRf>
*SRE?
```

<NRf>	0 to 255

Details

The Service Request Enable register determines which registers of the status byte register can generate service requests.

The query returns the bit sum of the Service Request Enable register.

Example

```
*SRE 48
*SRE?
```

The command sets the ESB and MAV bits of the Service Request Enable register, where:

- ESB (bit B5) = 32
- MAV (bit B4) = 16

If the service request enable register is 00100000, the return is 32.

Also see

None

*STB?

This command reads the Status Byte register.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*STB?
```

Details

Returns the contents of the Status Byte register using the Master Summary Status (MSS) bit (bit 6).

Example

*STB?	Queries the status byte. Example return if the Status Byte register contains the binary value 01100000: 96

Also see

None

*TRG

This command sends a bus trigger to the 2260B.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*TRG
```

Details

Use the `*TRG` command to issue a GPIB trigger to the 2260B. It has the same effect as a group execute trigger (`GET`).

If the 2260B cannot accept a trigger at the time of the command, the error message −211, "Trigger ignored," is generated.

Example

`*TRG`	Generates a trigger event.

Also see

None

*TST?

This command runs self test and reads the result.

Type	Affected by	Where saved	Default value
Query only	Not applicable	Not applicable	Not applicable

Usage

```
*TST?
```

Details

This command returns `0` if the self test completes with no errors.

If another value is returned, the self test detected an error.

Example

`*TST?`	Initiates the self-test.

Also see

None

*WAI

This command postpones the execution of subsequent commands until all previous overlapped commands are finished.

Type	Affected by	Where saved	Default value
Command only	Not applicable	Not applicable	Not applicable

Usage

```
*WAI
```

Details

The `*WAI` command halts further command processing until all present commands are complete.

Example

`*WAI`	Stop command processing until all commands are finished.

Also see

None

Section 6

Status model

In this section:

Status register introduction

The status model consists of status register sets and queues. You can monitor the status model to view instrument events and configure the status model to control the events.

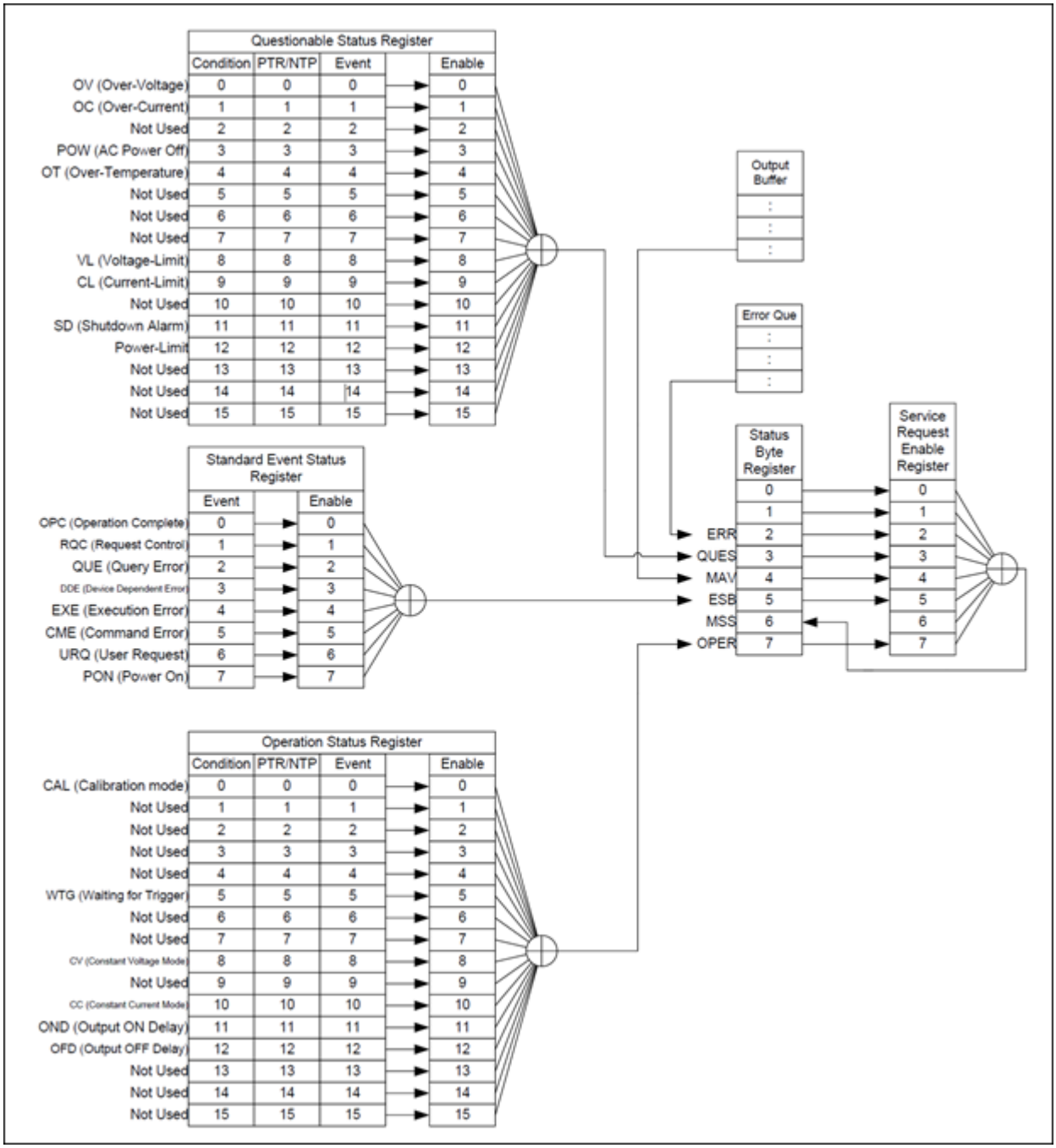
As you work with the status model, be aware that the result applies to the Status Byte Register. All the status register sets and queues flow into the Status Byte Register. Your test program can read this register to determine if a service request (SRQ) has occurred, and if so, which event caused it. The status registers maintain the status of the protection conditions, operation conditions and instrument errors.

The Status Byte Register, register sets, and queues include:

- Standard Event Status Register
- Questionable Status Register
- Operation Status Register
- Service Request Enable Register
- Service Request Generation
- Output Buffer
- Error Queue

The following figure shows the structure of the status model.

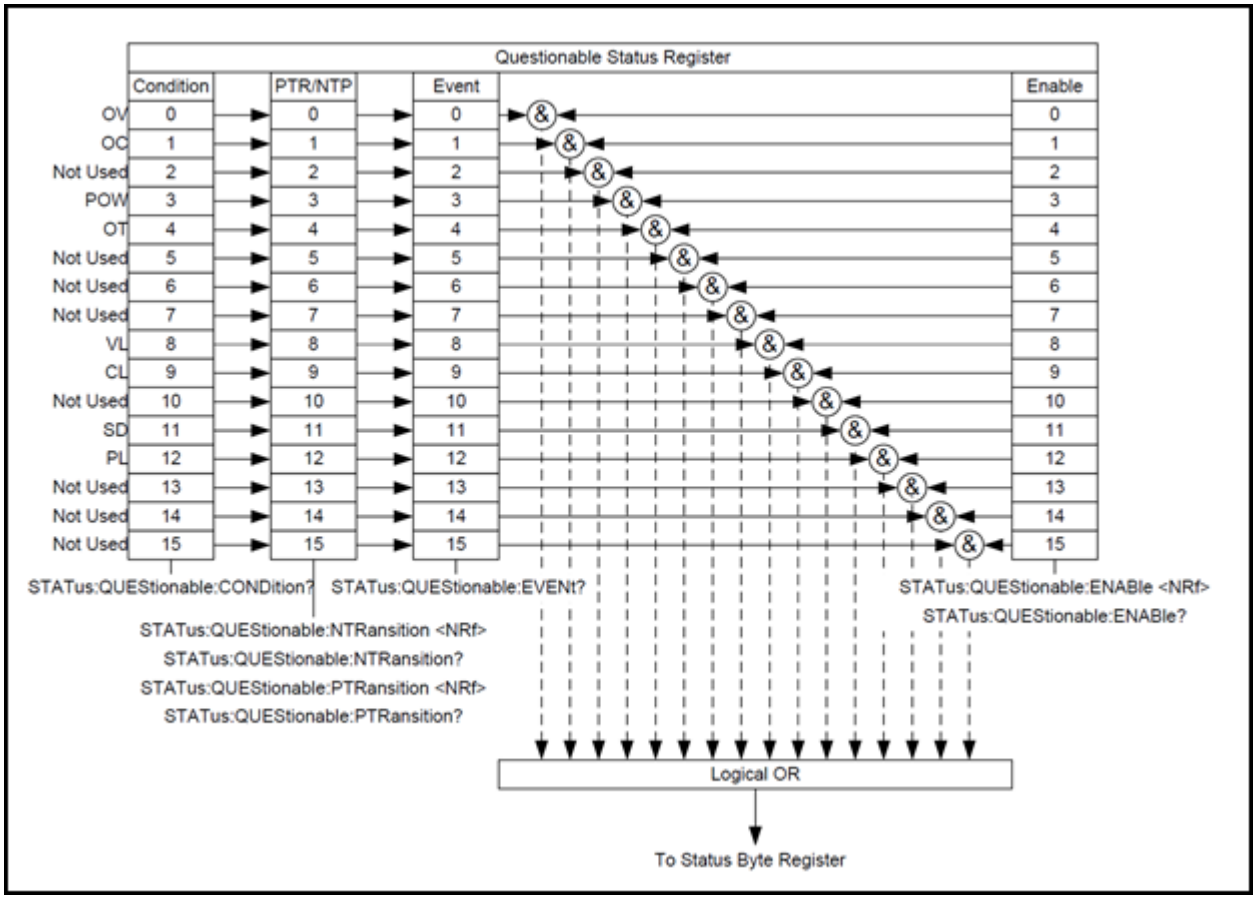
Figure 22: 2260B Status Model overview



Questionable Status Register

The Questionable Status Register indicates if any protection modes or limits have been tripped.

Figure 23: 2260B Questionable Status Register



Bit summary

Event	Bit number	Bit weight
OV (overvoltage) Overvoltage protection was tripped	0	1
OC (overcurrent) Overcurrent protection was tripped	1	2
POW (AC power off) AC power switch is off	3	8
OT (overtemperature) Overtemperature protection was tripped	4	16
VL (voltage limit) Voltage limit was reached	8	256
CL (current limit) Current limit has been reached	9	512
SD (shutdown alarm)	11	2048
PL (power limit)	12	4096

The Questionable Status Condition Register indicates the status of the power supply. If a bit is set in the Condition register, it indicates that the event is true. Reading the condition register does not change the state of the condition register.

The positive and negative transition (PTR and NTR) filters determine the type of transition conditions that set the corresponding bit in the Event Registers. Use the positive transition filter to view events that change from false to positive. Use the negative transition filter to view events that change from positive to negative.

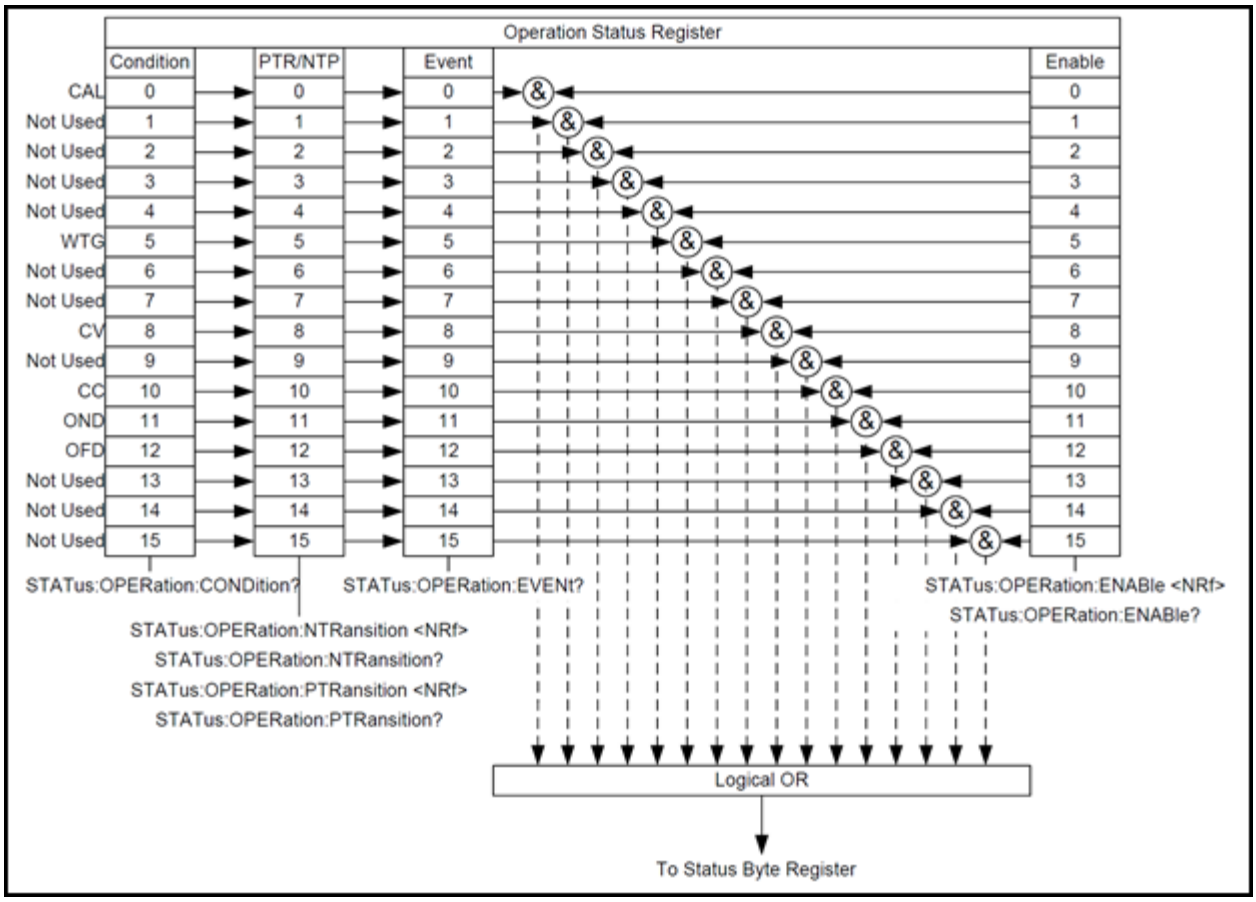
The PTR and NTR registers dictate the type of transition conditions that set the corresponding bits in the Event Register. If the Event Register is read, it is cleared to 0.

The Enable Register determines which events in the Event Register are used to set the QUES bit in the Status Byte Register.

Operation Status Register

The Operation Status Register indicates the operating status of the power supply.

Figure 24: 2260B Operation Status Register



Bit summary

Event	Bit number	Bit weight
CAL (calibration mode) Indicates if the 2260B is in calibration mode	0	1
WTG (waiting for trigger) Indicates if the 2260B is waiting for a trigger	5	32
CV (constant voltage mode) Indicates if the 2260B is in CV mode.	8	256
CC (constant current mode) Indicates if the 2260B is in CC mode.	10	1024
OND (output on delay) Indicates if output on delay time is active	11	2048
OFD (output off delay) Indicates if output off delay time is active	12	4096

The Operation Status Condition Register indicates the operating status of the power supply. If a bit is set in the Condition register, it indicates that the event is true. Reading the condition register does not change the state of the condition register.

The positive and negative transition (PTR and NTR) filters determine the type of transition conditions that set the corresponding bit in the Event Registers. Use the positive transition filter to view events that change from false to positive. Use the negative transition filter to view events that change from positive to negative.

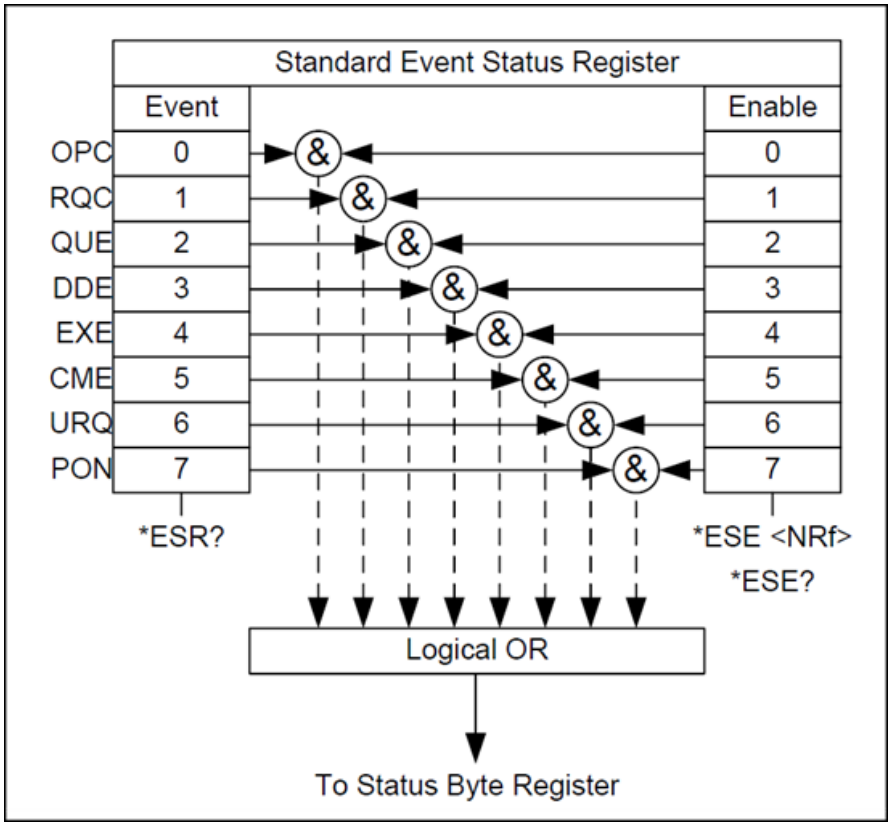
The PTR and NTR registers dictate the type of transition conditions that set the corresponding bits in the Event Register. If the Event Register is read, it is cleared to 0.

The Enable Register determines which events in the Event Register are used to set the OPER bit in the Status Byte Register.

Standard Event Status Register

The Standard Event Status Register indicates if any errors have occurred. The bits of the Event register are set by the error event queue.

Figure 25: 2260B Standard Event Status Register



Bit summary

Event	Bit number	Bit weight
OPC (operation complete) The OCP bit is set when all selected pending operations are complete. This bit is set in response to the `*OPC` command	0	1
RQC (request control)	1	2
QUE (query error) The query error bit is set in response to an error reading the output queue. This can be caused by trying to read the output queue when there is no data present.	2	4
DDE (device dependent error) Device specific error.	3	8
EXE (execution error) The EXE bit indicates an execution error due to one of the following: Illegal command parameter, parameter out of range, invalid parameter, or the command did not execute due to an overriding operation condition.	4	16
CME (command error) The CME bit is set when a syntax error has occurred. The CME bit can also be set when a `<GET>` command is received within a program message.	5	32
URQ (user request)	6	64
PON (power on) Indicates the power is turned on.	7	128

Any bits set in the event register indicate that an error has occurred. Reading the Event register resets the register to 0.

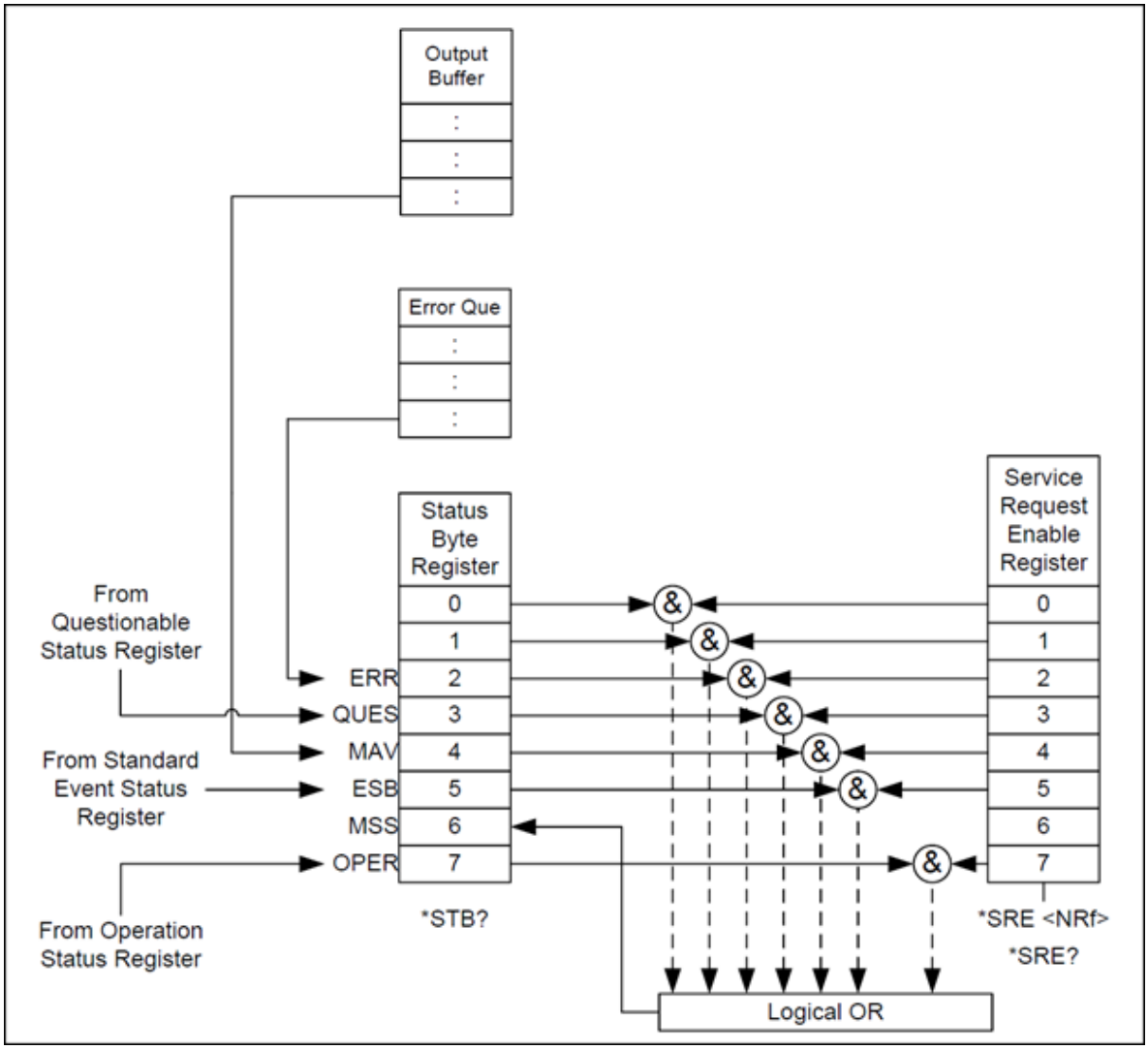
The Enable register determines which Events in the Event Register are used to set the ESB bit in the Status Byte Register.

Status Byte Register and Service Request Enable Register

The Status Byte register consolidates the status events of all status registers. You can read the Status Byte register with the `*STB?` query and can clear it with the `*CLS` command. Any bits set in the Status byte register act as a summary register for the other status registers and indicate if there is a service request, an error in the Error Queue, or data in the Output Queue. Reading the Status Byte register resets the register to 0.

The Service Request Enable Register controls which bits in the Status Byte Register can generate service requests.

Figure 26: 2260B Status Byte Register



Bit summary

Event	Bit number	Bit weight
ERR (error event/queue) If data is present in the error queue, the ERR bit is set.	2	4
QUES (questionable status register) The summary bit for the Questionable Status Register.	3	8
MAV (message available) This is set when there is data in the Output Queue waiting to be read.	4	16
(ESB) event summary bit The ESB is the summary bit for the Standard Event Status Register.	5	32
MSS bit The MSS bit is the summary of the Status Byte Register and Service Request Register (bits 1 to 5 and 7). This is set to 1.	6	64
OPER (operation status register) The OPER bit is the summary bit for the Operation Status Register.	7	128

Section 7

Error list

In this section:

Command errors

An error or event number in the range of −199 to 100 indicates that an IEEE 488.2 syntax error was detected by the parser of the instrument.

The occurrence of any error in this class causes the command error bit (bit 5) in the event status register (IEEE 488.2, section 11.5.1) to be set. One of the following events has occurred:

- An IEEE 488.2 syntax error was detected by the parser. In other words, a controller-to-device message was received that is in violation of the IEEE 488.2 standard. Possible violations include a data element that violates the device listening formats or whose type is unacceptable to the device.
- An unrecognized header was received. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEEE 488.2 common commands.

Events that generate command errors shall not generate execution errors, device-specific errors, or query errors. Refer to Execution errors (on page 7-4), Device-specific errors (on page 7-6), and Query errors (on page 7-6) for information on those errors.

–100 Command Error

This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that a Command Error, as defined in IEEE 488.2,11.5.1.1.4, has occurred.

–102 Syntax error

An unrecognized command or data type was encountered. For example, a string was received when the device does not accept strings.

–103 Invalid separator

The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit such as `MEAS:VOLT:DC?:MEASCURR:DC?`

–104 Data type error

The parser recognized a data element different than one allowed. For example, numeric or string data was expected, but block data was encountered.

–108 Parameter not allowed

More parameters were received than expected for the header. For example, the `KLOCk` command only accepts one parameter, so receiving `SYSTem:KLOCk 1,0` is not allowed.

–109 Missing parameter

Fewer parameters were received than required for the header. For example, the `KLOCk` command requires one parameter, so receiving `KLOCk` is not allowed.

–111 Header separator error

A character that is not a legal header separator was encountered while parsing the header. For example, no blank space followed the header, so `APPL 5, 1` is an error.

–112 Program mnemonic too long

The header contains more than twelve characters (see IEEE 488.2, 7.6.1.4.1).

–113 Undefined header

The header is syntactically correct, but it is undefined for this specific device. For example, `*XYZ` is not defined for any device.

–114 Header suffix out of range

The value of a numeric suffix attached to a program mnemonic makes the header invalid. See SCPI *Volume 1: Syntax and Style* section 6.2.5.2.

–115 Unexpected number of parameters

The number of parameters received does not correspond to the number of parameters expected. This is typically due an inconsistency with the number of instruments in the selected group.

–120 Numeric data error

This error and errors –121 through –129 are generated when parsing a data element which appears to be numeric, including the nondecimal numeric types. This error message is used if the device cannot detect a more specific error.

–121 Invalid character in number

An invalid character for the data type being parsed was encountered. For example, an alphabetic character in a decimal numeric or a `9` in octal data.

–128 Numeric data not allowed

A legal numeric data element was received, but the device does not accept one in this position for the header.

–131 Invalid suffix

The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.

–141 Invalid character data

Either the character data element contains an invalid character or the element received is not valid for the header.

–148 Character data not allowed

A legal character data element was encountered where prohibited by the device.

–151 Invalid string data

A string data element was expected, but was invalid (see IEEE 488.2, 7.7.5.2). For example, an `END` message was received before the terminal quote character.

–158 String data not allowed

A string data element was encountered but was not allowed by the device at this point in parsing.

–160 Block data error

This error and errors −161 through −169 are generated when parsing a block data element. The −160 error message is used if the device cannot detect a more specific error.

–161 Invalid block data

A block data element was expected, but was invalid (see IEEE 488.2, 7.7.6.2). For example, an `END` message was received before the length was satisfied.

–168 Block data not allowed

A legal block data element was encountered but was not allowed by the device at this point in parsing.

–178 Expression data not allowed

A legal expression data was encountered but was not allowed by the device at this point in parsing.

Execution errors

An error or event number in the range −299 to −200 indicates that an error has been detected by the execution control block of the instrument.

The occurrence of any error in this class causes the execution error bit (bit 4) in the event status register (IEEE 488.2, section 11.5.1) to be set. One of the following events has occurred:

- A `<PROGRAM DATA>` element following a header was evaluated by the device as outside of its legal input range or is otherwise inconsistent with the capabilities of the device.
- A valid program message could not be properly executed due to some device condition.

Execution errors are reported by the device after rounding and expression evaluation operations have taken place. Rounding a numeric data element, for example, is not reported as an execution error. Events that generate execution errors do not generate Command errors (on page 7-1), Device-specific errors (on page 7-6), or Query errors (on page 7-6).

–200 Execution error

This is the generic syntax error for devices that cannot detect more specific errors. This code indicates only that an Execution Error as defined in IEEE 488.2, 11.5.1.1.5, has occurred.

–201 Invalid while in local

Indicates that a command is not executable while the device is in local due to a hard local control (see IEEE 488.2, 5.6.1.5). For example, a device with a rotary switch receives a message that would change the switches state, but the device is in local so the message cannot be executed.

–203 Command protected

Indicates that a legal password-protected program command or query could not be executed because the command was disabled.

–211 Trigger ignored

Indicates that a `GET`, `*TRG`, or triggering signal was received and recognized by the device but was ignored because of device timing considerations. For example, the device was not ready to respond.

A DT0 device always ignores `GET` and treats `*TRG` as a command error.

–213 Init ignored

Indicates that a request for a measurement initiation was ignored because another measurement was already in progress.

–220 Parameter error

Indicates that a program data element related error occurred. This error message occurs when the device cannot detect the more specific errors described for errors −221 through −229.

–221 Settings conflict

Indicates that a legal program data element was parsed but could not be executed due to the present device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5.).

–222 Data out of range

Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5.).

–224 Illegal parameter value

Used where an exact value from a list of possible values was expected.

Device-specific errors

An error or event number in the range −399 to −300 or 1 to 32767 indicates that the instrument has detected an error that is not a command error, a query error, or an execution error. When this error or event occurs, some device operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class causes the device-specific error bit (bit 3) in the event status register (IEEE 488.2, section 11.5.1) to be set.

Events that generate device-specific errors do not generate command errors (on page 7-1), execution errors (on page 7-4), or query errors (on page 7-6).

–310 System error

Indicates that some error, termed "system error" by the device, has occurred. This code is device-dependent.

–320 Storage fault

Indicates that the firmware detected a fault when using data storage. This error is not an indication of physical damage or failure of any mass storage element.

Query errors

An `<error/event number>` in the range [−499 , −400] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol described in IEEE 488.2, chapter 6. The occurrence of any error in this class causes the query error bit (bit 2) in the event status register (IEEE 488.2, section 11.5.1) to be set. These errors correspond to message exchange protocol errors described in IEEE 488.2, section 6.5. One of the following is true:

- An attempt is being made to read data from the output queue when no output is either present or pending.
- Data in the output queue has been lost.

Events that generate query errors do not generate command errors (on page 7-1), execution errors (on page 7-4), or device-specific errors (on page 7-6).

–400 Query error

This is the generic query error for devices that cannot detect more specific errors. This code indicates only that a Query Error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.

Error messages and messages

The following error messages or messages may appear on the 2260B screen during operation.

Error messages	Description
Err 001	USB Mass Storage is not present
Err 002	No (such)file in USB mass storage
Err 003	Empty memory location
Err 004	File access error
Note	For error messages other than Err 001 to Err 004, please contact Keithley for service repair.
Messages	**Description**
MSG 001	External control of output. Output off (F-94=0, High=on)
MSG 002	External control of output. Output off (F-94=1, Low=on)
MSG 003	F-93 is not zero. Unable to calibrate.

Keithley Instruments • 28775 Aurora Road • Cleveland, Ohio 44139 • 1-800-833-9200 • tek.com/keithley

KEITHLEY
A Tektronix Company

04/2022