



**2 Series Mixed Signal Oscilloscope
MSO22, MSO24**

Programmer Manual





**2 Series Mixed Signal Oscilloscope
MSO22, MSO24
Programmer Manual**

Supports FW version 1.42.5 and above
www.tek.com

077-1776-01

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

TekSecure is a trademark of Tektronix, Inc.

2 Series MSO Programmer Manual. Released April 2022.

Contacting Tektronix

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

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

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

Table of Contents

Preface	iii
Getting Started	1-1
Setting Up Remote Communications Hardware	1-1
Ethernet	1-1
USB.....	1-1
Setting Up Remote Communications Software	1-2
Using VISA.....	1-2
Using a Socket Server	1-3
Documentation	1-5
Dynamic programmatic interface	1-5
Implicit activation	1-5
Creating, deleting, and listing dynamic instances	1-6
Backwards Compatibility	1-6
Command Syntax	2-1
Command and Query Structure	2-1
Clearing the instrument output queue	2-3
Command Entry.....	2-3
Constructed Mnemonics	2-5
Argument Types.....	2-7
Command groups	2-11
Acquisition command group	2-11
Act On Event command group	2-11
AFG Command Group	2-13
Alias command group	2-14
Battery command group.....	2-14
Bus command group.....	2-15
Calibration command group	2-19
Callouts command group	2-20
Cursor command group	2-21
Digital command group	2-26
Display control command group	2-27
Ethernet Command Group	2-30
File system command group.....	2-31
Horizontal command group.....	2-33
Mask command group.....	2-34
Math command group	2-35
Measurement command group	2-37
Miscellaneous command group	2-47
Pattern Generator group.....	2-50

Plot command group	2-51
Save and Recall command group	2-52
Save On command Group	2-53
Search and Mark command group	2-54
Self Test command group	2-60
Status and Error command group	2-61
Trigger command group	2-62
Vertical command group	2-66
Waveform Transfer command group	2-68
Zoom command group	2-72
Commands listed in alphabetical order.....	2-73
Status and Events	3-1
Registers	3-1
Queues	3-4
Event Handling Sequence.....	3-5
Synchronization Methods.....	3-5
Appendix A: Character Set	A-1
Appendix B: Reserved Words.....	B-1
Appendix C: Factory Defaults	C-1
Default Setup.....	C-1
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples	D-1
Example 1: Analog Waveform (Channels 1–4)	D-1
Example 2: Digital Waveform	D-3
Example 3: Digital with 4 Bytes Per Point and Zoom Off.....	D-5
Example 4: Digital with 8 Bytes Per Point and Zoom Off.....	D-7
Example 5: Digital with 4 Bytes Per Point and Zoom On	D-10
Example 6: RF Frequency Domain Waveform.....	D-12
Example 7: Baseband IQ Waveform	D-14
Appendix E: Search and Trigger Command Sequence Examples.....	E-1
Example 1: Single Threshold Edge Search	E-1
Example 2: Single Threshold Edge Trigger	E-1
Example 3: Dual Threshold Runt Search	E-2
Example 4: Single Threshold Logic Search on Three Waveforms.....	E-3
Glossary	
Index	

Preface

This programmer guide provides you with the information required to use Programmable Interface commands to remotely control your instrument.

The programmer manual is divided into the following major topics:

- **Getting started.** This topic introduces you to the programming information and provides basic information about setting up your instrument for remote control.
- **Command syntax.** This topic provides an overview of the command syntax that you use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.
- **Commands listed in alphabetical order.** This topic contains all commands listed in alphabetical order. Command details, syntax, and examples are provided.
- **Status and events.** This topic discusses the status and event reporting system for the programming interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument might return, including error messages.
- **Appendices.** These topics contain miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and interface specifications that can be helpful when using commands to remotely control the instrument.

Getting Started

This manual explains the use of commands to remotely control your instrument. With this information, you can write computer programs to perform functions, such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs.

Familiarity with the User Manual for your instrument is assumed. You can download the User Manual from the Tektronix website at www.tek.com.

NOTE. Most examples in this document assume that both *HEADer* and *VERBose* are set to *ON*.

Setting Up Remote Communications Hardware

You can remotely control communications between your instrument and a PC via Ethernet or USB cables.

Ethernet If you are using Ethernet, start by connecting an appropriate Ethernet cable to the Ethernet port (RJ-45 connector) on the side of your instrument. This connects the instrument to a 10BASE-T/100BASE-T/1000BASE-T local area network.

To change the Ethernet settings on your instrument, do the following:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Select the **LAN** panel.
4. In the menu, if you are on a DHCP Ethernet network that supplies the IP address automatically by a DHCP, tap **Auto**.
5. In the menu, if you want to supply your own network settings, tap **Manual** to set a hard coded TCP/IP address.

USB If you are using USB, start by connecting the appropriate USB cable to the USB 2.0 device port on the side of your instrument. This port requires that the cable connected from the port to the host computer meets the USB 2.0 specification. Typically, such cables should be 3 feet or shorter in length, but this is determined by the quality of the cable and, with higher quality cables, this length can be extended. (It is also dependent upon the drive capability of the host USB port to which the instrument is connected.) The use of high quality short cables is recommended to avoid USB connection problems.

With USB, the system automatically configures itself. To verify that the USB is enabled:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Touch **USB Device Port** to open the USB Device Port configuration menu.
4. If USB is disabled, tap **USB Device Port** to enable the USB Device port.

After connection, the host, with appropriate software, can list the instrument as a USB device with the following parameters: (See Table 1-1.)

Table 1-1: USB Device Parameters

Parameter	Value
Manufacturer ID	0x0699 (decimal 1689)
Product ID	0x105 (You can send the USBTMC:PRODUCTID:HEXadecimal? query to read the value)
Serial number	Serial number
Manufacturer description	"Tektronix"
Interface description	"USBTMC-USB488"

Setting Up Remote Communications Software

Connect your instrument directly to a computer to let the PC analyze your data, collect screen images, or to control the instrument using a program of your own creation. Three ways to connect your instrument to a computer are through the VISA drivers, or via a socket server.

Using VISA

VISA lets you use your MS-Windows computer to acquire data from your instrument for use in an analysis package that runs on your PC, such as Microsoft Excel, National Instruments LabVIEW, Tektronix OpenChoice Desktop software, or your own custom software. You can use a common communications connection, such as USB or Ethernet, to connect the computer to the instrument.

To set up VISA communications between your instrument and a computer:

1. Load the VISA drivers on your computer. Also, load your application, such as OpenChoice Desktop. You will find the drivers and OpenChoice Desktop software on the appropriate CD that comes with your instrument or at the Tektronix software finder Web page (www.tektronix.com/downloads).
2. Connect the instrument to your computer with the appropriate USB or Ethernet cable. Cycle the power on the instrument.
3. Select the **Utility** drop-down menu
4. Select **I/O** menu.
5. If you are using USB, the system sets itself up automatically for you, if USB is enabled. Check **USB DEVICE PORT** panel to be sure that USB is enabled. If it is not enabled, toggle the On/Off button to On.
6. To use Ethernet, select the **LAN** panel. Use the controls to adjust your network settings, as needed. For more information, see the e*Scope setup information below.
7. If you want to change socket server parameters, select the **Socket Server** panel and enter new values through the resulting panel controls.
8. Run your application software on your computer.

Quick Tips

- The USB 2.0 device port is the correct USB port for computer connectivity. Use the side USB 2.0 host ports to connect your instrument to USB flash drives, hard drives, keyboards or mice. Use the USB Device port to connect your instrument to a PC.

There are two USB 2.0 host ports and a USB 2.0 device port on the side of the instrument. Printers are not supported on host ports. PictBridge printers are not supported on the device port.

Using a Socket Server

A socket server provides two-way communication over an Internet Protocol-based computer network. You can use your instrument's socket server feature to let your instrument talk to a remote-terminal device or computer.

To set up and use a socket server between your instrument and a remote terminal or computer:

1. Connect the instrument to your computer network with an appropriate Ethernet cable.
2. Select the **Utility** drop-down menu.
3. Select the **I/O** menu.
4. Tap **Socket Server**.
5. On the resulting Socket Server panel, tap the top entry to toggle the Socket Server On.

6. Choose whether the protocol should be **None** or **Terminal**. A communication session run by a human at a keyboard typically uses a terminal protocol. An automated session might handle its own communications without using such a protocol.
7. If required, change the port number by rotating multipurpose knob **a**.
8. If required, press **OK** to set the new port number.
9. After setting up the socket server parameters, you can now have the computer “talk” to the instrument. If you are running an MS Windows PC, you could run its default client with its command-like interface. One way to do this is by typing “**Te**lnet ” in the Run window. The Telnet window will open on the PC.

NOTE. *On MS Windows 10, you must first enable Telnet in order for it to work.*

10. Start a terminal session between your computer and your instrument by typing in an open command with the instrument LAN address and port #. You can obtain the LAN address by pushing the **LAN** panel to view the resulting LAN setting panel. You can obtain the port # by tapping the **Socket Server** panel and viewing the **Port** item.

For example, if the instrument IP address was 123.45.67.89 and the port # was the default of 4000, you could open a session by writing into the MS Windows Telnet screen:

```
open 123.45.67.89 4000
```

The instrument will send a help screen to the computer when it is done connecting.

11. You can now type in a standard query, as found in the programmer manual, such as *i dn?

The Telnet session window will respond by displaying a character string describing your instrument. You can type in more queries and view more results on this Telnet session window. You can find the syntax for relevant queries and related status codes in other sections of this manual.

NOTE. *Do not use the computer’s backspace key during an MS Windows’ Telnet session with the instrument.*

Socket Server Terminal Protocol Mode Commands. Following are Tektronix Instrument Control Terminal Session Control commands:

!t <timeout> : set the response timeout in milliseconds.

!d : send device clear to the instrument.

!r : read response from instrument.

!h : print this usage info.

NOTE. *Commands containing a ? are treated as queries, and the responses are read automatically.*

Documentation Documentation for your instrument is available for download at www.tek.com.

Dynamic programmatic interface

This programmatic interface is dynamic. This means the instrument will not recognize certain commands until the objects referenced by those commands actually exist. For example, commands related to measurements are not recognized until measurements are added. Therefore, the response to a *LRN? query will not normally include the instrument's complete command set.

The following command groups are not available when the instrument is in its default state:

- Measurement
- Math
- Bus
- Search and Mark
- Plot

Adding an instance of one of those components will cause all commands related to that component to be recognized. For example, sending the [MEASUREMENT:ADDNew](#) command adds a measurement at which point the measurement commands will be recognized. Conversely, once all instances of a component have been deleted, the commands related to that component will no longer be recognized.

Implicit activation

When you send a command or query related to a dynamic object (such as Math1, Bus3, or Meas2) to the instrument and that instance does not yet exist, the instrument:

- creates a default instance with the name you specified
- adds all relevant commands and queries to the set of recognized commands
- responds to the command or query

Example 1 *RST followed by a *LRN? will not return any MATH<x> commands because in the default state, the instrument does not have any math waveforms. However querying :MATH:MATH1:DEFinition? will add MATH1 with the default math expression Ch1 - Ch2. Then the query will return the expected result. Note that if Ch1 or Ch2 is not active, they will be activated as part of this action. A *LRN issued after this will return MATH commands in addition to other available commands.

Example 2 *RST followed by :MEASUREMENT:MEAS3:TYPE? creates a measurement named MEAS3 and returns its type. Since the default type is Period, you will get Period as the response. A *LRN issued after this will return all MEASUREMENT:MEAS3 commands in addition to other available commands.

Not all commands in these groups start implicit activation. ADDNew, DELEte, and LIST commands do not result in implicit activation.

Creating, deleting, and listing dynamic instances

You can create a new default instance of a dynamic feature by using the ADDNew command. For example, :MEASUREMENT:ADDNew "Meas1" will create a new measurement named Meas1. Meas1 will be a Period measurement since the default type for measurements is Period. Note that you can change Meas1 type to any other supported type using :MEASUREMENT:MEAS1:TYPE command. The DELEte command deletes the named dynamic instance. For example :BUS:DELEte "B2" will delete a bus named B2 if it exists. You can delete only one instance at a time. The LIST command returns a list of all dynamic instances currently in existence. For example, if you have added three measurements named Meas1, Meas2 and Meas3, :MEASUREMENT:LIST? returns MEAS1, MEAS2, MEAS3.

Backwards Compatibility

2 Series MSO instruments have implicit backwards compatibility built into the command set. Many commands from previous scope generations can be successfully processed by modern scopes. In particular, the MSO/DPO5000/7000/70000 command set has been partially migrated to the new generation. Use the following table to check if a legacy command is supported and how it will be interpreted by a 2 Series MSO.

The left column indicates the legacy command and the right column indicates the command that will be executed on a 2 Series MSO. Some functions on the scope have been simplified, so multiple legacy commands are mapped to a single modern command.

In addition to these implicit aliases, the 2 Series MSO also contain a PI Translator feature. This allows users to define custom command processing to improve

compatibility with legacy automation code. Please refer to the Programming Interface Translator Technical Brief on tek.com for more information.

Legacy oscilloscope command	2 Series MSO command alias
:BUS:B<n>:CAN:BITRate:VALue	:BUS:B<n>:CAN:BITRate:CUSTom
:BUS:B<n>:CAN:FD:BITRate:VALue	:BUS:B<n>:CAN:FD:BITRate:CUSTom
:MEASUrement:MEAS<n>:EDGEQUALifier	:MEASUrement:MEAS<n>:SOUrce<n>
:MEASUrement:MEAS<n>:ACQTime	:MEASUrement:MEAS<n>:WAITTime
:DPOJET:GATing	:MEASUrement:MEAS<n>:GATing
:BUS:B<n>:I2C:SCLk:SOUrce	:BUS:B<n>:I2C:CLOCK:SOUrce
:BUS:B<n>:I2C:ADDRes:RWINClude	:BUS:B<n>:I2C:DATa:SOUrce
:MEASUrement:MEAS<n>:FUNDAMENTALFreq	:MEASUrement:MEAS<n>:FREQ
:BUS:B<n>:LIN:VALue	:BUS:B<n>:LIN:BITRate:CUSTom
:MATH:MATH<n>:SPECTral:UNWRap	:MATH:MATH<n>:SPECTral:WRAP:STATE
:MATH:MATH<n>:SPECTral:UNWRap:DEGrees	:MATH:MATH<n>:SPECTral:WRAP:DEGrees
:DPOJET:INTERp	:MEASUrement:INTERp
:BUS:B<n>:RS232C:TX:SOUrce	:BUS:B<n>:RS232C:SOUrce
:BUS:B<n>:RS232C:TX:SOUrce:THReshold	:BUS:B<n>:RS232C:SOUrce:THReshold
:BUS:B<n>:SPI:SCLk:SOUrce	:BUS:B<n>:SPI:CLOCK:SOUrce
:BUS:B<n>:SPI:MOSi:INPut	:BUS:B<n>:SPI:DATa:SOUrce
:BUS:B<n>:SPI:SS:SOUrce	:BUS:B<n>:SPI:SElect:SOUrce
:BUS:B<n>:SPI:MOSi:THReshold	:BUS:B<n>:SPI:DATa:THReshold
:BUS:B<n>:SPI:SCLk:POLarity	:BUS:B<n>:SPI:CLOCK:POLarity
:BUS:B<n>:SPI:MOSi:DATa:POLarity	:BUS:B<n>:SPI:DATa:POLarity
:BUS:B<n>:SPI:SS:POLarity	:BUS:B<n>:SPI:SElect:POLarity
:TRIGger:A:BUS:B<n>:SPI:DATa:IN:VALue	:TRIGger:A:BUS:B<n>:SPI:DATa:VALue
:TRIGger:A:BUS:B<n>:SPI:DATa:MISO:VALue	
:TRIGger:A:BUS:B<n>:SPI:DATa:OUT:VALue	
:TRIGger:A:BUS:B<n>:SPI:DATa:MOSI:VALue"	
:MEASUrement:MEAS<n>:DELay:DIREction	:MEASUrement:MEAS<n>:TOEDGESEARCHDIRec
:MATH:MATH<n>:POSITION	:DISPlay:WAVEView<n>:MATH:MATH<n>:VERTical:POSition
:MATH:MATH<n>:SCALE	:DISPlay:WAVEView<n>:MATH:MATH<n>:VERTical:SCALE
:REF:REF<n>:POSITION	:DISPlay:WAVEView<n>:REF:REF<n>:VERTical:POSition
:REF:REF<n>:SCALE	:DISPlay:WAVEView<n>:REF:REF<n>:VERTical:SCALE
CH<x>:YUNits	CH<x>:PROBEFunc:EXTUnits

:CURSor:VBArS:POSITIONA	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:APOSition
:CURSor:VBArS:POSA"	
:CURSor:VBArS:POSITIONB	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:BPOSition
:CURSor:VBArS:POSB"	
:CURSor:HBArS:POSITIONA	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:APOSition
:CURSor:HBArS:POSITIONB	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:BPOSition
:CURSor:SCREEN:XPOSITIONA	:DISPlay:WAVEView<n>:CURSor:CURSOR:SCREEN:AXPOSition
:CURSor:SCREEN:XPOSITIONB	:DISPlay:WAVEView<n>:CURSor:CURSOR:SCREEN:BXPOSition
:CURSor:SCREEN:YPOSITIONA	:DISPlay:WAVEView<n>:CURSor:CURSOR:SCREEN:AYPOSition
:CURSor:SCREEN:YPOSITIONB	:DISPlay:WAVEView<n>:CURSor:CURSOR:SCREEN:BYPOSition
:CURSor:WAVEform:POSITIONA	:DISPlay:WAVEView<n>:CURSor:CURSOR:WAVEform:APOSition
:CURSor:WAVEform:POSITIONB	:DISPlay:WAVEView<n>:CURSor:CURSOR:WAVEform:BPOSition
:CURSor:SOURCEA	:DISPlay:WAVEView<n>:CURSor:CURSOR:ASOURce
:CURSor:SOURCEB	:DISPlay:WAVEView<n>:CURSor:CURSOR:BSOURce
:CURSor:VBArS:UNItS	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:UNItS
:CURSor:HBArS:UNITA	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:AUNItS
:CURSor:HBArS:UNITA	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:AUNItS
:CURSor:HBArS:UNItS	
:CURSor:HBArS:UNITB	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:BUNItS
:CURSor:VBArS:ALTERNATEA	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:ALTERNATEA
:CURSor:VBArS:ALTERNATEB	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:ALTERNATEB
:CURSor:VBArS:DELTA	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:DELTA
:CURSor:WAVEform:HDELTA"	:DISPlay:WAVEView<n>:CURSor:CURSOR:HBArS:DELTA
:CURSor:HBArS:DELTA	
:CURSor:WAVEform:VDELTA"	:DISPlay:WAVEView<n>:CURSor:CURSOR:VBArS:DELTA
:CURSor:VBArS:DELTA	
:CURSor:DDT	:DISPlay:WAVEView<n>:CURSor:CURSOR:DDT
:CURSor:ONEOVERDELTAVALUE	:DISPlay:WAVEView<n>:CURSor:CURSOR:ONEOVERDELTAVALUE
:CURSor:FUNCTION	:DISPlay:WAVEView<n>:CURSor:CURSOR:FUNCTION
:CURSor:STATE	:DISPlay:WAVEView<n>:CURSor:CURSOR:STATE
:CURSor:CURSOR:STATE"	
:CURSor:MODE	:DISPlay:WAVEView<n>:CURSor:CURSOR:MODE
:BUS:B<n>:POSition	:DISPlay:WAVEView<n>:BUS:B<n>:VERTical:POSition

:ZOOM:HORIZONTAL:POSITION	:DISPLAY:WAVEVIEW<n>:ZOOM:ZOOM<n>:HORIZONTAL:POSITION
:ZOOM:ZOOM<n>:HORIZONTAL:POSITION"	
:ZOOM:HORIZONTAL:SCALE	:DISPLAY:WAVEVIEW<n>:ZOOM:ZOOM<n>:HORIZONTAL:SCALE
:ZOOM:ZOOM<n>:HORIZONTAL:SCALE"	
:ZOOM:VERTICAL:POSITION	:DISPLAY:WAVEVIEW<n>:ZOOM:ZOOM<n>:VERTICAL:POSITION
:ZOOM:ZOOM<n>:VERTICAL:POSITION"	
:ZOOM:VERTICAL:SCALE	:DISPLAY:WAVEVIEW<n>:ZOOM:ZOOM<n>:VERTICAL:SCALE
:ZOOM:ZOOM<n>:VERTICAL:SCALE"	
:ZOOM:STATE	:DISPLAY:WAVEVIEW<n>:ZOOM:ZOOM<n>:STATE
:ZOOM:ZOOM<n>:STATE"	
:BUS:B<n>:STATE	:DISPLAY:GLOBAL:B<n>:STATE
CH1:STATE	DISPLAY:GLOBAL:CH1:STATE
:MATH:MATH<n>:STATE	:DISPLAY:GLOBAL:MATH<n>:STATE
:REF:REF<n>:STATE	:DISPLAY:GLOBAL:REF<n>:STATE
:REM:REM<n>:STATE	:DISPLAY:GLOBAL:REM<n>:STATE
:PLOT:PLOT<n>:STATE	:DISPLAY:GLOBAL:PLOT<n>:STATE
:DISPLAY:INTENSITY:WAVEFORM(:ANALYSIS):FASTACQ)	:DISPLAY:WAVEVIEW<n>:INTENSITY:WAVEFORM
:DISPLAY:INTENSITY:GRATICULE	:DISPLAY:WAVEVIEW<n>:INTENSITY:GRATICULE
:DISPLAY:STYLE	:DISPLAY:WAVEVIEW<n>:STYLE
:DISPLAY:GRATICULE	:DISPLAY:WAVEVIEW<n>:GRATICULE
:DISPLAY:FILTER	:DISPLAY:WAVEVIEW<n>:FILTER
:DISPLAY:VIEWSTYLE	:DISPLAY:WAVEVIEW<n>:VIEWSTYLE
:HORIZONTAL:DELAY:POSITION	:HORIZONTAL:POSITION
:HORIZONTAL:MAIN:POSITION"	
:HORIZONTAL:DELAY:POSITION	
:HORIZONTAL:MODE:SCALE	:HORIZONTAL:SCALE
:HORIZONTAL:SECdiv	
:HORIZONTAL:MAIN:SCALE	
:HORIZONTAL:MAIN:SECdiv	
:HORIZONTAL:DELAY:SCALE	
:HORIZONTAL:DELAY:SECdiv"	
:HORIZONTAL:MAIN:UNITS:STRING	:HORIZONTAL:MAIN:UNITS
:HORIZONTAL:DELAY:STATE	:HORIZONTAL:DELAY:MODE
:HORIZONTAL:MAIN:DELAY:MODE"	

:HORizontal:DELay:TiMe(:RUNSAfter :TRIGAAfter)	:HORizontal:DELay:TiMe
:HORizontal:MAIn:DELay:TiMe"	
:CH1:VOLTS	CH1:SCALe
:HORizontal:MODE:SAMPLERate	:HORizontal:SAMPLERate
:HORizontal:MAIn:SAMPLERate	
:HORizontal:DiGital:SAMPLERate	
:HORizontal:DiGital:SAMPLERate:MAIn"	
:HORizontal:MODE:RECOrdlength	:HORizontal:RECOrdlength
:HORizontal:RESOLution	
:HORizontal:DiGital:RECOrdlength	
:HORizontal:DiGital:RECOrdlength:MAIn	
:HORizontal:ACQLENGTH"	
:CH<X>:AMPSViolts:ENABLE	:CH<X>:PROBEFunc:EXTUnits:STATE
:CH<X>:AMPSViolts:FACTor	:CH<X>:SCALERATIo
:SAVEON:WAVEform:FILEFormat	:SAVEONEVent:WAVEform:FILEFormat
:SAVEON:WAVEform:SOURce	:SAVEONEVent:WAVEform:SOURce
:SAVEON:IMAGe:FILEFormat	:SAVEONEVent:IMAGe:FILEFormat
:SAVEON:FILE:NAME	:SAVEONEVent:FILEName
:SAVEON:FILE:NAME	:SAVEONEVent:FILEName
:TRIGger:A:PULse:WIDth:WHEn	:TRIGger:A:PULSEWidth:WHEn
:TRIGger:A:PULse:WIDth:POLarity	:TRIGger:A:PULSEWidth:POLarity
:TRIGger:A:PULse:WIDth:HIGHLimit	:TRIGger:A:PULSEWidth:HIGHLimit
:TRIGger:A:PULse:WIDth:LOWLimit	:TRIGger:A:PULSEWidth:LOWLimit
:TRIGger:A:PULSEWidth:WIDth"	
:TRIGger:A:PULse:RUNT:WHEn	:TRIGger:A:RUNT:WHEn
:TRIGger:A:PULse:RUNT:POLarity	:TRIGger:A:RUNT:POLarity
:TRIGger:A:PULse:RUNT:WIDth	:TRIGger:A:RUNT:WIDth
:TRIGger:A:PULse:TIMEOut:POLarity	:TRIGger:A:TIMEOut:POLarity
:TRIGger:A:PULse:TIMEOut:TiMe	:TRIGger:A:TIMEOut:TiMe
:TRIGger:A:RISEFall:SOURce	:TRIGger:A:TRANSition:SOURce
:TRIGger:A:RISEFall:WHEn	:TRIGger:A:TRANSition:WHEn
:TRIGger:A:PULse:TRANSition:WHEn"	
:TRIGger:A:RISEFall:WHEn	:TRIGger:A:TRANSition:WHEn
:TRIGger:A:PULse:TRANSition:WHEn"	

:TRIGger:A:RISEFall:POLarity	:TRIGger:A:TRANSition:POLarity
:TRIGger:A:PULse:TRANSition:POLarity"	
:TRIGger:A:RISEFall:POLarity	:TRIGger:A:TRANSition:POLarity
:TRIGger:A:PULse:TRANSition:POLarity"	
:TRIGger:A:RISEFall:DELTatime	:TRIGger:A:TRANSition:DELTatime
:TRIGger:A:PULse:TRANSition:DELTatime"	
:TRIGger:A:RISEFall:DELTatime	:TRIGger:A:TRANSition:DELTatime
:TRIGger:A:PULse:TRANSition:DELTatime"	
:TRIGger:A:LOGIc:SETHold:CLOCK:SOURce	:TRIGger:A:SETHold:CLOCK:SOURce
:TRIGger:A:LOGIc:SETHold:CLOCK:EDGE	:TRIGger:A:SETHold:CLOCK:EDGE
:TRIGger:A:LOGIc:SETHold:SETTime	:TRIGger:A:SETHold:SETTime
:TRIGger:A:LOGIc:SETHold:HOLDTime	:TRIGger:A:SETHold:HOLDTime
:TRIGger:A:LOGIc:PATtern:WHEn	:TRIGger:A:LOGIc:WHEn
:TRIGger:A:LOGIc:PATtern:DELTatime	:TRIGger:A:LOGIc:DELTatime
:TRIGger:A:LOGIc:STATE:CLOCK:SOURce	:TRIGger:A:LOGIc:INPut:CLOCK:SOURce
:TRIGger:A:LOGIc:INPut:CLOCK:EDGE	:TRIGger:A:LOGIc:POLarity
:TRIGger:A:BUS:B<n>:CAN:ADDRess:MODE	:TRIGger:A:BUS:B<n>:CAN:IDentifier:MODE
:TRIGger:A:BUS:B<n>:CAN:ADDRess:VALue	:TRIGger:A:BUS:B<n>:CAN:IDentifier:VALue
:TRIGger:A:BUS:B<n>:PARAllel:VALue	:TRIGger:A:BUS:B<n>:PARAllel:DATa:VALue
:TRIGger:A:BUS:B<n>:RS232C:RX:DATa:VALue	:TRIGger:A:BUS:B<n>:RS232C:DATa:VALue
:TRIGger:A:BUS:B<n>:RS232C:RX:DATa:VALue	:TRIGger:A:BUS:B<n>:RS232C:DATa:VALue
:TRIGger:A:BUS:B<n>:RS232C:RX:DATa:SIZE	:TRIGger:A:BUS:B<n>:RS232C:DATa:SIZE

Command Syntax

You can control the operations and functions of the instrument through the Ethernet port or the USB 2.0 device port using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

Backus-Naur Form Notation

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

Table 2-1: Symbols for Backus-Naur Form

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

Command and Query Structure

Commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and status information.

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark at the end. For example, the set command `ACQUIRE:MODE` has a query form `ACQUIRE:MODE?`. Not all commands have both a set and a query form. Some commands have set only and some have query only.

Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, defined in the following table.

Table 2-2: Command Message Elements

Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

Commands Commands cause the instrument to perform a specific function or change one of the settings. Commands have the structure:

[:] <Header> [<Space> <Argument> [<Comma> <Argument>] . . .]

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

Queries Queries cause the instrument to return status or setting information. Queries have the structure:

- [:] <Header>
- [:] <Header> [<Space> <Argument> [<Comma> <Argument>] . . .]

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

Headers Use the `HEADer` command to control whether the instrument returns headers as part of the query response. If header is on, the query response returns command headers, then formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

Table 2-3: Comparison of Header Off and Header On Responses

Query	Header Off	Header On
TIME?	"14:30:00"	:TIME "14:30:00"
ACQuire:NUMAVg?	100	:ACQUIRE:NUMAVG 100

Clearing the instrument output queue

You can clear the Output Queue and reset the instrument a new command or query by using the selected Device Clear (DCL) function.

Command Entry

The following rules apply when entering commands:

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

Abbreviating You can abbreviate many instrument commands. Each command in this documentation shows the minimum acceptable abbreviations in capitals. For example, you can enter the command `ACQuire:NUMAVg` simply as `ACQ:NUMA` or `acq:numa`.

Abbreviation rules may change over time as new instrument models are introduced. Thus, for the most robust code, use the full spelling.

If you use the `HEADer` command to have command headers included as part of query responses, you can further control whether the returned headers are abbreviated or are full-length with the `VERBoSe` command.

Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated commands in the order received.

When concatenating commands and queries, you must follow these rules:

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:MODE NORMAl` and `ACQuire:NUMAVg 8`, can be concatenated into the following single command:

```
TRIGger:MODE NORMAl;:ACQuire:NUMAVg 8
```

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands `ACQuire:MODE ENVeLope` and `ACQuire:NUMAVg 8` into a single command:

```
ACQuire:MODE ENVeLope; NUMAVg 8
```

The longer version works equally well:

```
ACQuire:MODE ENVeLope;:ACQuire:NUMAVg 8
```

3. Never precede a star (*) command with a colon:

```
ACQuire:STATE 1;*OPC
```

Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODE ENVeLope;*OPC;NUMAVg 8` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 8.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display graticule is set to Full and the display style is set to dotsonly, the concatenated query `DISPlay:GRATiCuLe?;STYLe?` will return the following.

If the header is on:

```
DISPLAY:GRATICULE FULL;:DISPLAY:STYLE DOTSONLY
```

If the header is off:

FULL;DOTSONLY

1. Set commands and queries may be concatenated in the same message. For example,

```
ACQuire:MODE SAMple;NUMAVg?;STATE?
```

is a valid message that sets the acquisition mode to sample. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

Here are some invalid concatenations:

```
DISPlay:STYle DOTsonly OFF;ACQuire:NUMAVg 8 (no colon before ACQuire)
```

```
DISPlay:GRAticule FULL;:STYle DOTSONLY OFF (extra colon before STYle.
```

```
DISPlay:GRAticule FULL;:*TRG (colon before a star (*) command)
```

Terminating This documentation uses <EOM> (End of Message) to represent a message terminator.

Table 2-4: End of Message Terminator

Symbol	Meaning
<EOM>	Message terminator

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII line feed (LF) character.

This instrument does not support ASCII LF only message termination. The instrument always terminates outgoing messages with LF and EOI.

Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, CH4, CH5, CH6, CH7, or CH8 depending on the number of FlexChannels in your instrument. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:POSition command, and there is also a CH2:POSition command. In the command descriptions, this list of choices is abbreviated as CH<x>.

Bus Mnemonics Commands specify the bus to use as a mnemonic in the header.

Table 2-5: Bus Mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥ 1 .

Channel Mnemonics

Commands specify the channel to use as a mnemonic in the header.

Table 2-6: Channel Mnemonics

Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

Cursor Position Mnemonics

When cursors are displayed, commands may specify which cursor of the pair to use.

Table 2-7: Cursor Mnemonics

Symbol	Meaning
CURSOR<x>	A cursor selector; <x> is must be 1 or 2.

Math Specifier Mnemonics

Commands can specify the mathematical waveform to use as a mnemonic in the header.

Table 2-8: Math Specifier Mnemonics

Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is ≥ 1 .

Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header.

Table 2-9: Measurement Specifier Mnemonics

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is ≥ 1 .

Reference Waveform Mnemonics

Commands can specify the reference waveform to use as a mnemonic in the header.

Table 2-10: Reference Waveform Mnemonics

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is ≥ 1 .

View Mnemonics Commands can specify the view to use as a mnemonic in the header.

Table 2-11: Waveview Mnemonics

Symbol	Meaning
WAVEView<x>	A waveview specifier; <x> must be equal to 1.
PLOTView<x>	A plotview specifier; <x> must be equal to 1.
MATHFFTView<x>	A mathfftview specifier; <x> must be equal to 1.

Search Mnemonics Commands can specify a search to use as a mnemonic in the header.

Table 2-12: Search Mnemonics

Symbol	Meaning
SEARCH<x>	A Search specifier; <x> is ≥ 1 .

Zoom Mnemonics Commands can specify a zoom to use as a mnemonic in the header.

Table 2-13: Zoom Mnemonics

Symbol	Meaning
ZOOM<x>	A zoom specifier; <x> must be equal to 1.

Argument Types

Commands use arguments such as enumeration, numeric, quoted string and block. Each of these arguments are listed in detail below.

Enumeration Enter these arguments as unquoted text words. Like key words, enumeration arguments follow the same convention where the portion indicated in uppercase is required and that in lowercase is optional.

For example: ACQuire:MODe SAMPlE

Numeric Many instrument commands require numeric arguments. The syntax shows the format that the instrument returns in response to a query. This is also the preferred format when sending the command to the instrument, though any of the formats will be accepted. This documentation represents these arguments as described below.

Table 2-14: Numeric Arguments

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent
<bin>	Signed or unsigned integer in binary format

Most numeric arguments will be automatically forced to a valid setting, by either rounding or truncating, when an invalid number is input, unless otherwise noted in the command description.

Quoted String

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: "This is a quoted string". This documentation represents these arguments as follows:

Table 2-15: Quoted String Argument

Symbol	Meaning
<QString>	Quoted string of ASCII text

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: "this is a valid string".
2. You can mix quotation marks within a string as long as you follow the previous rule. For example: "this is an 'acceptable' string".
3. You can include a quote character within a string by repeating the quote. For example: "here is a "" mark".
4. Strings can have upper or lower case characters.
5. A carriage return or line feed embedded in a quoted string does not terminate the string. The return is treated as another character in the string.
6. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some invalid strings:

- "Invalid string argument' (quotes are not of the same type)
- "test<EOI>" (termination character is embedded in the string)

Block

Some commands use a block argument form to define a range or type of value, as defined in the table below.

Table 2-16: Block Argument

Symbol	Meaning
<NZDig>	A nonzero digit character in the range of 1–9
<Dig>	A digit character, in the range of 0–9
<DChar>	A character with the hexadecimal equivalent of 00 through FF (0 through 255 decimal)
<Block>	A block of data bytes defined as: <Block> ::= {#<NZDig><Dig>[<Dig>...][<DChar>...]}#0[<DChar>...]<terminator>

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <NZDig> and <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

Command groups

The programmable interface conforms to Tektronix standard codes and formats except where noted. The interface also conforms to IEEE Std 488.2-1987 except where noted.

Acquisition command group

Acquisition commands set up the modes and functions that control how the instrument acquires signals and processes them into waveforms. Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

Table 2-17: Acquisition commands

Command	Description
ACquire?	Queries the current acquisition state.
ACquire:MAXSamplerate?	Returns the maximum real-time sample rate.
ACquire:MODE	Sets or queries the acquisition mode.
ACquire:NUMACq?	Returns the number of waveform acquisitions that have occurred since starting acquisition with the <code>ACquire:STATE RUN</code> command.
ACquire:NUMAVg	Sets or queries number of acquisitions for an averaged waveform.
ACquire:SEquence:CURrent?	In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.
ACquire:SEquence:MODE	In single sequence acquisition, the single sequence stop after count is based on number of acquisitions or measurements. Number of acquisitions is the only mode supported for this product.
ACquire:SEquence:NUMSEquence	In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence.
ACquire:STATE	Starts, stops, or returns acquisition state.
ACquire:STOPAfter	Sets or queries whether the acquisition is continuous or single sequence.

Act On Event command group

Use this group of commands to program the instrument to perform an action on trigger, search, measurement limit, and mask test events.

Table 2-18: Act On Event commands

Command	Description
ACTONEVent:ENable	Enables or disables actions on event (AOE).
ACTONEVent:LIMit	Sets whether the act on event should limit the number of saves, so as not to fill the hard drive.
ACTONEVent:LIMITCount	If the number of act on event saves is limited, this command sets the limit.
ACTONEVent:MASKFail:ACTION:SAVEIMAGE:STATE	Save a screen capture when a mask test fails.
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask test fails.
ACTONEVent:MASKFail:ACTION:SRQ:STATE	Generates an SRQ event when a mask fails.
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE	Stops acquisitions when a mask test fails.
ACTONEVent:MASKHit:ACTION:SAVEIMAGE:STATE	Saves a screen capture when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SRQ:STATE	Generates an SRQ event when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE	Stops acquisitions when a mask hit occurs.
ACTONEVent:MASKPass:ACTION:SAVEIMAGE:STATE	Saves a screen capture when a mask test passes.
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask test passes.
ACTONEVent:MASKPass:ACTION:SRQ:STATE	Generates an SRQ event when a mask passes
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE	Stops acquisitions when a mask test passes.
ACTONEVent:MEASUrement:ACTION:SAVEIMAGE:STATE	Saves a screen capture when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SRQ:STATE	Generates an SRQ event when any measurement triggers the user-defined measurement limits.
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE	Stops acquisitions when the user-set measurement limit is exceeded.
ACTONEVent:SEARCH:ACTION:SAVEIMAGE:STATE	Saves a screen capture when a search event is found.
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a search event is found.
ACTONEVent:SEARCH:ACTION:SRQ:STATE	Generates an SRQ event when any search event is found.

Table 2-18: Act On Event commands (cont.)

Command	Description
ACTONEvent:SEARCH:ACTION:STOPACQ:STATE	Stops acquisitions when a search event is found.
ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE	Saves a screen capture on a trigger event from a single sequence or sequence of N acquisition.
ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) on a trigger event from a single sequence or sequence of N acquisition.
ACTONEvent:TRIGger:ACTION:SRQ:STATE	Generates an SRQ event when a trigger event occurs.
ACTONEvent:TRIGger:ACTION:STOPACQ:STATE	Stops acquisitions on a trigger event from a single sequence or sequence of N acquisition.
SAVEONEvent:FILEDest	Sets or queries the file path.
SAVEONEvent:FILEName	Sets or queries the file name without the extension.
SAVEONEvent:IMAGe:FILEFormat	Sets or returns the file extension (png, jpg, bmp).
SAVEONEvent:WAVEform:FILEFormat	Sets or returns the file extension (csv, wfm, mat).
SAVEONEvent:WAVEform:SOUrce	Sets or returns the sources for saving waveforms when an event occurs.

AFG Command Group

Use the AFG commands for Arbitrary Function Generator functionality. Requires option AFG.

Table 2-19: AFG commands

Command	Description
AFG:AMPLitude	Sets (or queries) the AFG amplitude in volts, peak to peak.
AFG:ARBitrary:SOUrce	Sets or queries the source name for the Arbitrary Waveform.
AFG:BURSt:CCOUnt	Sets or returns the cycle count for AFG burst mode.
AFG:BURSt:TRIGger	Triggers a burst on AFG output.
AFG:FREQuency	Sets (or queries) the AFG frequency, in Hz.
AFG:FUNction	Sets (or queries) which AFG function to execute.
AFG:HIGHLevel	Sets (or queries) the high level value of the output waveform, in volts.
AFG:LOWLevel	Sets (or queries) the low level value of the output waveform, in volts.
AFG:NOISEAdd:PERCent	Sets (or queries) the AFG additive noise level as a percentage.
AFG:NOISEAdd:STATE	Sets (or queries) the AFG additive noise state.
AFG:OFFSet	Sets (or queries) the AFG offset value, in volts.
AFG:OUTPut:LOAd:IMPEDance	Sets (or queries) the AFG output load impedance.
AFG:OUTPut:MODE	Sets or returns the AFG output mode.
AFG:OUTPut:STATE	Sets (or queries) the AFG output state.
AFG:PERIod	Sets (or queries) the period of the AFG waveform, in seconds.
AFG:PULse:WIDTh	Sets (or queries) the AFG pulse width, in seconds.

Table 2-19: AFG commands (cont.)

Command	Description
AFG:RAMP:SYMmetry	Sets (or queries) the AFG ramp symmetry as a percentage.
AFG:SQUare:DUty	Sets (or queries) the AFG duty cycle, as a percentage.

Alias command group

Alias commands allow you to define new commands as a sequence of standard commands. You might find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program date, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered errors unless specifically noted in the command description in this document.

Table 2-20: Alias commands

Command	Description
ALlas	Sets or queries the alias state.
ALlas:CATalog?	Returns a list of the currently defined alias labels.
ALlas:DEFine	Assigns a sequence of program messages to an alias label.
ALlas:DELEte	Removes a specified alias.
ALlas:DELEte:ALL	Deletes all existing aliases.
ALlas:DELEte:NAME	Removes a specified alias.
ALlas:STATE	Sets or queries the alias state.

Battery command group

Battery commands gather battery information from your instrument.

Table 2-21: Battery commands

Command	Description
BATTery:ACPOWER?	Queries the state of AC being plugged in.
BATTery:SLOT<1,2>:CHARGE?	Queries the current charge of the battery.
BATTery:SLOT<1,2>:INSTALLED?	Queries if a battery is installed.
BATTery:SLOT<1,2>:SERIALnumber?	Queries the serial number of the battery.
BATTery:SLOT<1,2>:TIMETOEMPTY?	Queries the time to empty of the battery.
BATTery:SLOT<1,2>:TIMETOFULL?	Queries the time to full of the battery.

Bus command group

Use the commands in the Bus Command Group to configure a bus. These commands let you:

- Specify the bus type.
- Specify the signals to be used in the bus.
- Specify its display style.

NOTE. *Bus commands are present once a bus has been added.*

Bus Mnemonics Commands specify the bus to use as a mnemonic in the header.

Table 2-22: Bus mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥ 1 .

Table 2-23: Bus commands

Command	Description
BUS:ADDNew	Adds the specified bus.
BUS:B<x>:CAN:BITRate	Sets or queries the CAN bit rate.
BUS:B<x>:CAN:BITRate:VALue	Sets or queries CAN bit rate.
BUS:B<x>:CAN:FD:BITRate	Sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:FD:BITRate:CUSTom	Sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:SAMPLEpoint	Sets or queries the sample point for the specified CAN bus.
BUS:B<x>:CAN:SIGNAL	Sets or queries the signal type for the specified CAN bus.
BUS:B<x>:CAN:SOURce	Sets or queries the CAN source channel.

Table 2-23: Bus commands (cont.)

Command	Description
BUS:B<x>:CAN:STANdard	Sets or queries which CAN standard specification to analyze the specified CAN bus with.
BUS:B<x>:CAN:THReshold	Sets or queries the source channel threshold for the specified CAN bus.
BUS:B<x>:DISplay:FORMat	Sets or queries how the data is represented in the busform for the specified bus.
BUS:B<x>:DISplay:LAYout	This command sets or queries the format a bus layer should use.
BUS:B<x>:I2C:CLOCK:SOUrce	Sets or queries the I2C clock (SCLK) source for the specified bus.
BUS:B<x>:I2C:CLOCK:THReshold	Sets or queries the I2C Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:I2C:DATa:SOUrce	Sets or queries the I2C data (SDA) source for the specified bus.
BUS:B<x>:I2C:DATa:THReshold	Sets or queries the I2C Data (SDA) source threshold for the specified bus.
BUS:B<x>:I2C:RWINADDR	Determines whether decoded I2C slave addresses are pure seven-bit values, or have the R/W* combined with them.
BUS:B<x>:LABel:COLor	Sets or queries the color of the specified bus label.
BUS:B<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified bus label.
BUS:B<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified bus label.
BUS:B<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified bus label.
BUS:B<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified bus label, such as Arial or Times New Roman.
BUS:B<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified bus label.
BUS:B<x>:LABel:name	Sets or queries the waveform label for the specified bus.
BUS:B<x>:LABel:XPOS	Sets or queries the x-position of the specified bus' label.
BUS:B<x>:LABel:YPOS	Sets or queries the y-position of the specified bus' label.
BUS:B<x>:LIN:BITRate	Sets or queries the LIN bus bit rate.
BUS:B<x>:LIN:IDFORMat	Sets or queries LIN bus id format.
BUS:B<x>:LIN:POLarity	Sets or queries the LIN bus polarity.
BUS:B<x>:LIN:SAMPLEpoint	Specifies the point to sample during each bit period, as a percent, for the specified LIN bus.
BUS:B<x>:LIN:SOUrce	Sets or queries sets the LIN bus source.
BUS:B<x>:LIN:SOUrce:THReshold	Sets or queries the LIN source threshold for the specified bus.
BUS:B<x>:LIN:STANdard	Sets or queries the LIN bus standard.
BUS:B<x>:LIN:BITRate:CUSTom	Sets or queries LIN custom bit rate for the specified bus.
BUS:B<x>:PARAllel:ALLTHResholds	Sets or queries the threshold for all sources for the parallel bus.
BUS:B<x>:PARAllel:ALLTHResholds:APPLY	Sets all of the data source thresholds to the value of the allMRefs parameter for the parallel bus.
BUS:B<x>:PARAllel:BIT<x>SOUrce	Sets or queries the specified bit source for specified parallel bus.
BUS:B<x>:PARAllel:BIT<n>SOUrce:THReshold	Sets or queries the specified bit source threshold for the specified parallel bus.
BUS:B<x>:PARAllel:CLOCK:EDGE	Determines which edges of its clock signal cause a clocked parallel bus to sample new states.
BUS:B<x>:PARAllel:CLOCK:ISCLOCKED	Determines whether the bus operates in a clocked or asynchronous fashion.

Table 2-23: Bus commands (cont.)

Command	Description
BUS:B<x>:PARAllel:CLOCKSOURCE	Sets or queries the Parallel clock source for the specified bus.
BUS:B<x>:PARAllel:CLOCKSOURCE:THRESHOLD	Sets or queries the bit source threshold for the parallel bus.
BUS:B<x>:RS232C:BITRate	Sets or queries the RS-232 bit rate for the specified bus.
BUS:B<x>:RS232C:BITRate:CUSTOM	Sets or queries the RS232 custom bit rate for the specified bus.
BUS:B<x>:RS232C:DATABits	Sets or queries the number of RS-232 data bits for the specified bus.
BUS:B<x>:RS232C:DELIMITER	Sets or queries the RS-232 delimiting value for a packet on the specified bus.
BUS:B<x>:RS232C:DISPLAYMODE	Sets or queries the display mode for the specified bus.
BUS:B<x>:RS232C:PARITY	Sets or queries the RS-232 parity for the specified bus.
BUS:B<x>:RS232C:POLARITY	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOURCE	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOURCE:THRESHOLD	Sets or queries the RS232 source threshold for the specified bus.
BUS:B<x>:SENT:CHANWidth	Sets or queries SENT fast channel bit widths for the specified bus.
BUS:B<x>:SENT:NIBBLECount	Sets or queries SENT data nibbles for the specified bus.
BUS:B<x>:SENT:NUMCHANNELS	Sets or queries SENT fast data channels for the specified bus.
BUS:B<x>:SENT:PAUSEPULSE	Sets or queries SENT pause pulse for the specified bus.
BUS:B<x>:SENT:POLARITY	Sets or queries SENT Idle State signal polarity for the specified bus.
BUS:B<x>:SENT:SLOW	Sets or queries the SENT slow channel configuration for the specified bus.
BUS:B<x>:SENT:SOURCE	Sets or queries the SENT DATA source for the specified bus.
BUS:B<x>:SENT:THRESHOLD	Sets or queries the SENT DATA source threshold for the specified bus.
BUS:B<x>:SENT:TICKTIME	Sets or queries the SENT bus Clock Tick parameter for the specified bus.
BUS:B<x>:SENT:TICKTOLERANCE	Sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.
BUS:B<x>:SPI:BITOrder	Sets or queries the shift direction used to de-serialize data for the SPI mode of the bus.
BUS:B<x>:SPI:CLOCK:POLARITY	Sets or queries the SPI clock (SCLK) polarity for the specified bus.
BUS:B<x>:SPI:CLOCK:SOURCE	Sets or queries the SPI clock (SCLK) source for the specified bus.
BUS:B<x>:SPI:CLOCK:THRESHOLD	Sets or queries the SPI Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:SPI:DATA:POLARITY	Sets or queries the SPI data (DATA) polarity for the specified bus.
BUS:B<x>:SPI:DATA:SIZE	Sets or queries the number of bits per word for the specified bus.
BUS:B<x>:SPI:DATA:SOURCE	Sets or queries the SPI data (DATA) source for the specified bus.
BUS:B<x>:SPI:DATA:THRESHOLD	Sets or queries the SPI Data (DATA) source threshold for the specified bus.
BUS:B<x>:SPI:FRAMING	Sets or queries the SPI bus framing.
BUS:B<x>:SPI:IDLETime	Sets or queries the SPI bus idle time.
BUS:B<x>:SPI:MISO:DATA:POLARITY	Sets or queries the SPI MISO Data source polarity for the specified bus.
BUS:B<x>:SPI:MISO:INPUT	Sets or queries the SPI MISO source for the specified bus.
BUS:B<x>:SPI:MISO:THRESHOLD	Sets or queries the SPI MISO Data source threshold for the specified bus.
BUS:B<x>:SPI:MOSI:DATA:POLARITY	Sets or queries the SPI MOSI source polarity for the specified bus.

Table 2-23: Bus commands (cont.)

Command	Description
BUS:B<x>:SPI:MOSi:INPut	Sets or queries the SPI MOSI source for the specified bus.
BUS:B<x>:SPI:MOSi:THReshold	Sets or queries the SPI MOSI source threshold for the specified bus.
BUS:B<x>:SPI:NUMBer:INputs	Sets or queries the number of inputs for the specified bus.
BUS:B<x>:SPI:SELEct:POLarity	Sets or queries the SPI Slave Select (SS) polarity for the specified bus.
BUS:B<x>:SPI:SELEct:SOUrce	Sets or queries the SPI Slave Select (SS) source for the specified bus.
BUS:B<x>:SPI:SELEct:THReshold	Sets or queries the SPI Select (SS) source threshold for the specified bus.
BUS:B<x>:TYPE	Sets or queries the bus type specified.
BUS:DELEte	Deletes the specified bus.
BUS:LIST?	Lists all currently defined bus.
BUSTABLE:ADDNew	Adds the specified bus table.
BUSTABLE:DELEte	Deletes the specified bus table.
BUSTABLE:LIST?	Lists all currently defined bus tables.
DISPlay:WAVEView<x>:BUS:B<x>:STATE	Sets or queries the state of the specified bus in the specified Waveform View.
DISPlay:WAVEView<x>:BUS:B<x>:VERTical:POStion	Sets or queries the vertical position of the specified bus in the specified Waveform View.

Calibration command group

The Calibration commands provide information about the current state of instrument calibration and allow you to initiate signal path calibration (SPC).

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Table 2-24: Calibration commands

Command	Description
*CAL?	Instructs the instrument to perform signal path calibration and returns the calibration status when complete. Takes several minutes to run.
CALibrate?	Returns the calibration status.
CALibrate:INTERNAL	Starts the signal path calibration. Takes several minutes to run.
CALibrate:INTERNAL:START	Starts the signal path calibration.
CALibrate:INTERNAL:STATUS?	Returns the status of the signal path calibration.
CALibrate:PWRUpstatus?	Returns the current status of the power-up calibration.
TOUCHSCReen:STATE	Sets or queries the enabled state of the touch screen.

Callouts command group

The Callout commands creates custom callouts to document specific details of your test results.

Table 2-25: Callout commands

Command	Description
<code>CALLOUTS:CALLOUT<x></code>	Creates a new note.
<code>CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE</code>	Sets or queries the source of the Bookmark callout type.
<code>CALLOUTS:CALLOUT<x>:BOOKMark:XPOS</code>	Sets or queries the X-Position of the Bookmark callout type.
<code>CALLOUTS:CALLOUT<x>:COLOR</code>	Sets or queries the text color of the callout.
<code>CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:X</code>	Sets or queries horizontal display position of the callout text
<code>CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:Y</code>	Sets or queries vertical display position of the callout text
<code>CALLOUTS:CALLOUT<x>:FONT:BOLD</code>	Sets or queries the bold state of the callout text.
<code>CALLOUTS:CALLOUT<x>:FONT:ITALIC</code>	Sets or queries the italic state of the callout text.
<code>CALLOUTS:CALLOUT<x>:FONT:SIZE</code>	Sets or queries the font size of the callout text.
<code>CALLOUTS:CALLOUT<x>:FONT:TYPE</code>	Sets or queries the font type of the callout.
<code>CALLOUTS:CALLOUT<x>:FONT:UNDERLine</code>	Sets or queries the underline state of the callout text.
<code>CALLOUTS:CALLOUT<x>:TEXT</code>	Sets or queries the callout text.
<code>CALLOUTS:CALLOUT<x>:TYPE</code>	Sets or queries type of the callout

Cursor command group

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for each cursor, such as waveform source, and cursor position.

You can also use the commands to select one of the following cursor functions:

- **Off.** Shuts off the display of all cursors.
- **Vertical bars.** Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time). Vertical bars are another name for vertical screen cursors.
- **Horizontal bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them. Horizontal bars are another name for horizontal screen cursors.
- **Waveform cursors.** Consists of two cursors you can independently assign to a waveform. Waveform cursors enable you to conveniently measure waveform amplitude and time at specific points on the waveform. In XY format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen cursors.** Consist of two pairs of horizontal and vertical bar cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors are basically just turning on horizontal bars and vertical bars at the same time. These cursors have no association with any waveform, except that they inherit the color of the waveform they are assigned to.

NOTE. *Cursor commands are available once a view has been added.*

Table 2-26: Cursor commands

Command	Description
<code>DISplay:MATHFFTView<x>:CURSor:ASOUrce?</code>	Queries the cursor source for cursor A in the specified Math-FFT waveform.
<code>DISplay:MATHFFTView<x>:CURSor:BSOUrce?</code>	Queries the cursor source for cursor B in the specified Math-FFT waveform.
<code>DISplay:MATHFFTView<x>:CURSor:DDT?</code>	Queries the delta V over delta T cursor readout value in the specified Math-FFT waveform.
<code>DISplay:MATHFFTView<x>:CURSor:FUNctIon</code>	Sets or queries the cursor type in the specified Math-FFT waveform.
<code>DISplay:MATHFFTView<x>:CURSor:HBArs:APOsition</code>	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.

Table 2-26: Cursor commands (cont.)

Command	Description
DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOsition	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts?	Queries the cursor B vertical units in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:DELTA?	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:MODe	Sets or returns the cursor tracking mode in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTAVALUE?	Queries the one over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOsition	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOsition	Sets or returns the vertical cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOsition	Sets or returns the horizontal cursor B position of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOsition	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:STATE	Sets or queries the visible state of cursors in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArS:APOsition	Sets or queries the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArS:AUNIts?	Queries the vertical cursor A measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArS:BPOsition	Sets or queries the horizontal cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArS:BUNIts?	Queries the vertical cursor B measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArS:DELTA?	Queries the delta T cursor readout value of the specified cursor in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOsition	Sets or queries the waveform cursor A position in the specified plot view.
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOsition	Sets or queries the waveform cursor B position in the specified plot view.
DISplay:PLOTView<x>:CURSor:ASOURce?	Queries the cursor source for plot cursor A.
DISplay:PLOTView<x>:CURSor:BSOURce?	Queries the cursor source for plot cursor B.
DISplay:PLOTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:FUNCTion	Sets or queries the cursor mode of the specified cursor in the specified view.

Table 2-26: Cursor commands (cont.)

Command	Description
DISplay:PLOTView<x>:CURSor:HBArS:APOsition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArS:AUNIts?	Sets or queries the cursor A vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArS:BPOsition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArS:BUNIts?	Sets or queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArS:DELTA?	Sets or queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:MODE	Sets or queries the cursor tracking mode of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:ONEOVERDELTAVALUE?	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOsition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOsition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOsition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOsition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SPLITMODE	Sets or queries the cursor source mode in the specified view.
DISplay:PLOTView<x>:CURSor:STATE	Sets or queries the visible state of the cursor of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArS:APOsition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArS:BPOsition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArS:DELTA?	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArS:UNIts?	Queries the VBArS cursor readout units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:APOsition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:BPOsition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ASOURce?	Queries the cursor source for plot cursor A

Table 2-26: Cursor commands (cont.)

Command	Description
DISplay:REFFFTView<x>:CURSor:BSOURce?	Queries the cursor source for plot cursor B.
DISplay:REFFFTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:FUNCTion	Sets or queries the cursor type of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts?	Queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:DELTA?	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:MODE	Sets or queries the cursor tracking mode of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ONEOVERDELTAVALUE?	Queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SPLITMODE	Sets or queries whether both cursors have same or different source.
DISplay:REFFFTView<x>:CURSor:STATE	Sets or queries the visible state of the cursor of the specified cursor n the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:DELTA?	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:UNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition?	Queries the value of the cursor A horizontal position.

Table 2-26: Cursor commands (cont.)

Command	Description
DISplay:REFFFTView<x>:CURSor: WAVEform:APOSition	Sets or returns the waveform cursor A position in the specified plot view.
DISplay:REFFFTView<x>:CURSor: WAVEform:AVPOSition?	Queries the value of the cursor A vertical position.
DISplay:REFFFTView<x>:CURSor: WAVEform:BHPOSition?	Queries the value of the cursor B horizontal position.
DISplay:REFFFTView<x>:CURSor: WAVEform:BPOSition	Sets or returns the waveform cursor B position in the specified plot view.
DISplay:REFFFTView<x>:CURSor: WAVEform:BVPOSition?	Queries the value of the cursor B vertical position.
DISplay:WAVEView<x>:CURSor?	Queries the cursor parameters for the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>?	Queries the cursor parameters for the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:ASOUrce	Sets or queries the cursor A source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:BSOUrce	Sets or queries the cursor B source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:DDT?	Returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:FUNCTion	Sets or queries the cursor mode (SCREEN or DATA) of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:HBArS:APOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:HBArS:AUNIts?	Queries the cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:HBArS:BPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:HBArS:BUNIts?	Queries the cursor B vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:HBArS:DELTA?	Sets or queries the delta V cursor readout value over the history period.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:MODE	Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:ONEOVERDELTAVALUE?	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView:CURSor:CURSOR1: ROLOCATION	Sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).
DISplay:WAVEView<x>:CURSor: CURSOR<x>:SCREEN:AXPOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:SCREEN:AYPOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.

Table 2-26: Cursor commands (cont.)

Command	Description
DISplay:WAVEView<x>:CURSor: CURSOR<x>:SCREEN:BXPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:SCREEN:BYPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:SPLITMODE	Sets or queries the cursor source mode in the specified view.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:STATE	Sets or queries the visible state of the cursor of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:VBArS:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:VBArS:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:VBArS:DELTA?	Sets or queries the delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:VBArS:UNIts?	Queries cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:WAVEform:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR: WAVEform:AVPOSition?	Queries the vertical waveform value at the cursor A position in the specified Waveform View.
DISplay:WAVEView<x>:CURSor: CURSOR<x>:WAVEform:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR: WAVEform:BVPOSition?	Queries the vertical waveform value at the cursor B position in the specified Waveform View.

Digital command group

Use the commands in the Digital Command Group to acquire up to 16 digital signals and analyze them.

Table 2-27: Digital commands

Command	Description
DCH<x>:D<x>:THReshold	Sets or queries the threshold level in volts for the specified digital channel.
DCH<x>_D<x>:LABel:COLor	Sets or queries the color of the label of the specified digital bit.
DCH<x>_D<x>:LABel:FONT:BOLD	Sets or queries the bold state of the label of the specified digital bit.
DCH<x>_D<x>:LABel:FONT:ITALic	Sets or queries the italic state of the label of the specified digital bit.
DCH<x>_D<x>:LABel:FONT:SIZE	Sets or queries the font size of the label of the specified digital bit.
DCH<x>_D<x>:LABel:FONT:TYPE	Sets or queries the font type of the label of the specified digital bit.
DCH<x>_D<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the label of the specified digital bit.

Table 2-27: Digital commands (cont.)

Command	Description
DCH<x>_D<x>:LABel:NAME	Sets or queries the label name of the specified digital bit.
SElect:DCH<x>:DAI1	Turns on or off all constituent digital channels.

Display control command group

Display commands can be found in this section as well as the sections of related components. These commands control general instrument settings, such as the intensity of the graticule, stacked or overlay display mode, and the fastacq color palette. Display commands also control how and where waveforms are shown, their position on screen, and zoom settings applied to the view. For example, display commands can turn on or off the display of channels or set the selected source.

Some actions can create a new view which can have its own settings. For example, adding a histogram will create a new view where the Histogram plot is displayed. Each view acts as a separate window within the instrument application and can be rearranged or annotated as desired.

A WaveView is the primary view used for viewing inputs and time-domain signals. Buses, non-FFT maths, refs, and analog channels are displayed in the WaveView.

A PlotView is used for viewing measurement results and other plotted data. XY plots and FFTs are shown in individual PlotViews.

Each of these views can have separate settings, zoom, cursors, and annotations. Display commands which are view specific have a view parameter in the programmable interface, such as DISPLAY:WAVEView<x>: ... or DISPLAY:PLOTView<x>: ... Selected source can also be specified on a per-view basis. The overall selected source is determined by the selected view and the selected source within that view.

PlotViews and WaveViews have some differences in command syntax due to differences in view functionality. For example, WaveViews and PlotViews have a different zoom model. The commands for specifying the zoom reflects these differences, and different command syntax is available depending on the view.

Table 2-28: Display control commands

Command	Description
DISplay?	Returns current display settings.
DISplay:{CH<x> Math<x> REF<x>}: INVERTColor	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:{CH<x> Math<x> REF<x>}: NORMALColor	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:COLors	Sets or queries the color mode for the graticule and waveform display.

Table 2-28: Display control commands (cont.)

Command	Description
DISplay:GLObal:B<x>:STATE	Sets or queries the display mode (on or off) of the specified bus.
DISplay:GLObal:CH<x>:STATE	Sets or queries the display mode (on or off) of the specified channel (both analog and digital).
DISplay:GLObal:DCH<x>:STATE	Sets or queries the display mode (on or off) of the specified channel (digital).
DISplay:GLObal:MATH<x>:STATE	Sets or queries the display mode (on or off) of the specified math.
DISplay:GLObal:REF<x>:STATE	Sets or queries the display mode (on or off) of the specified reference.
DISplay:INTENSITY?	Returns the waveform and graticule saturation levels.
DISplay:INTENSITY:BACKLight	Sets or queries the waveform backlight intensity settings.
DISplay:INTENSITY:BACKLight:AUTODim:ENABle	Sets or queries the state of the display auto-dim feature.
DISplay:INTENSITY:BACKLight:AUTODim:TIME	Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.
DISplay:MATHFFTView<x>:AUTOScale	Sets or returns the enabled state of autoscale for plots.
DISplay:MATHFFTView<x>:GRIDlines	Sets or queries the grid lines setting of the plot.
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:MATHFFTView<x>:XAXIS:SCALE	Sets or queries the x-axis scale setting for FFT Math waveforms.
DISplay:MATHFFTView<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for FFT Maths.
DISplay:PERsistence	Sets or queries display persistence setting.
DISplay:PERsistence:RESET	Clears the persistence data.
DISplay:PLOTView<x>:AUTOScale	Sets or queries the enabled state of autoscale for plots.
DISplay:PLOTView<x>:GRIDlines	Sets or queries the Grid lines setting of the specified plot.
DISplay:PLOTView<x>:XAXIS:SCALE	Sets or queries the horizontal scale setting for applicable plot.
DISplay:PLOTView<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for applicable plots.
DISplay:REFFFTView<x>:AUTOScale	Sets or queries the enabled state of auto-scale for plots.
DISplay:REFFFTView<x>:GRIDlines	Sets or returns the grid lines setting of the plot.
DISplay:REFFFTView<x>:REF:REF<x>:STATE	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISplay:REFFFTView<x>:XAXIS:SCALE	Sets or queries the x-axis scale setting for REFFFT.
DISplay:SELEct:BUS	Sets or queries the overall selected bus.
DISplay:SELEct:MATH	Sets or queries the overall selected math.
DISplay:SELEct:REFerence	Sets or queries the overall selected reference waveform.
DISplay:SELEct:SOUrce	Sets or queries the overall selected source.
DISplay:SELEct:VIEW	Sets or queries the selected view.
DISplay:SELEct:WAVEView<x>:SOUrce	Sets or queries the selected source in the given waveview.
DISplay:VARpersist	Sets or queries the persistence decay time.
DISplay:WAVEform	Globally enables or disables the waveform display.
DISplay:WAVEView<x>:BUS:B<x>:STATE	Sets or queries the state of the specified bus in the specified Waveform View.

Table 2-28: Display control commands (cont.)

Command	Description
DISPlay:WAVEView<x>:BUS:B<x>:VERTical:POStion	Sets or queries the vertical position of the specified bus in the specified Waveform View.
DISPlay:WAVEView<x>:CH<x>:STATE	Sets or queries the state of the specified channel in the specified Waveform View.
DISPlay:WAVEView<x>:CH<x>:VERTical:POStion	Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions.
DISPlay:WAVEView<x>:CH<x>:VERTical:SCAlE	Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.
DISPlay:WAVEView<x>:DCH<x>_D<x>:STATE	Sets or queries the display state of the specified digital channel in the specified Waveform View.
DISPlay:WAVEView<x>:DCH<x>_DALL: DIGORDER	Sets or queries the order of the digital channels.
DISPlay:WAVEView<x>:DCH<x>_DALL: VERTical:POStion	Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions.
DISPlay:WAVEView<x>:FILTer	Sets or queries the type of interpolation filter for the display.
DISPlay:WAVEView<x>:GRATicule	Selects or queries the type of graticule that is displayed.
DISPlay:WAVEView<x>:INTENSITy: GRATicule	Sets or queries the graticule saturation level.
DISPlay:WAVEView<x>:INTENSITy: WAVEform	Sets or queries the waveform saturation level.
DISPlay:WAVEView<x>:MATH:MATH<x>: AUTOScale	Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.
DISPlay:WAVEView<x>:MATH:MATH<x>: STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISPlay:WAVEView<x>:MATH:MATH<x>: VERTical:POStion	Sets or queries the vertical position in divisions of the specified math waveform.
DISPlay:WAVEView<x>:MATH:MATH<x>: VERTical:SCAlE	Sets or queries the vertical scale of the specified math waveform.
DISPlay:WAVEView<y>:REF<x>_DALL: FRAMe	Sets or returns the selected frame of the specified digital ref.
DISPlay:WAVEView<y>:REF:REF<x>: FRAMe	sets or returns the selected frame of the specified analog ref.
DISPlay:WAVEView<x>:REF:REF<x>:STATE	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISPlay:WAVEView<x>:REF:REF<x>: VERTical:POStion	Sets or queries the vertical position in divisions of the specified reference waveform.
DISPlay:WAVEView<x>:REF:REF<x>: VERTical:SCAlE	Sets or queries the vertical scale of the specified reference waveform.
DISPlay:WAVEView<x>:STYle	Sets or queries the waveforms are displayed for analysis mode.
DISPlay:WAVEView<x>:VIEWStYle	Sets or queries the waveform layout style used by the display.
DISPlay:WAVEView:GRIDTYPE	Sets or queries the Waveform View Graticule mode (Fixed or Moveable).

Table 2-28: Display control commands (cont.)

Command	Description
DISPLAY:PLOTVIEW<x>:XAXIS:SCALE	Sets or queries the horizontal scale setting for applicable plots, either Linear or Log.
DISPLAY:PLOTVIEW<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for applicable plots, either Linear or Log.

Ethernet Command Group

Use the commands in the Ethernet Command Group to set up the 10BASE-T, 100BASE-TX, 1000BASE-TX or 100BASE-T Ethernet remote interface.

Table 2-29: Ethernet Commands

Command	Description
ETHERnet:DHCPbootp	Specifies the network initialization search for a DHCP/BOOTP server.
ETHERnet:DNS:IPADDRESS	Specifies the network Domain Name Server (DNS) IP address.
ETHERnet:DOMAINname	Specifies the network domain name.
ETHERnet:ENET:ADDRESS?	Returns the Ethernet address (MAC address) value assigned to the instrument.
ETHERnet:GATEWay:IPADDRESS	Specifies the network gateway IP address.
ETHERnet:IPADDRESS	Specifies the IP address assigned to the instrument.
ETHERnet:LXI:LAN:RESET	Resets the LXI local area network.
ETHERnet:LXI:LAN:SERVICENAME	Specifies the mDNS service name used for the LXI interface.
ETHERnet:LXI:LAN:STATUS?	Returns the LXI network status.
ETHERnet:NAME	Sets or queries the instrument Ethernet hostname assigned to the instrument.
ETHERnet:NETWORKCONFig	Specifies the Ethernet network configuration setting.
ETHERnet:PING	Causes the instrument to ping the gateway IP address.
ETHERnet:PING:STATUS?	Returns the results from sending the ETHERnet:PING command to ping the gateway IP address.
ETHERnet:SUBNETMask	Specifies the network subnet mask value.

File system command group

Use the commands in the File System Command Group to help you use the built-in hard disk drive. You can use the commands to do the following:

- List the contents of the current directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes:
“C:\MYDIR\TEK00001.SET”
- File names follow the MS-DOS format: [DRIVE:][\PATH\]filename
- Path separators can be either forward slashes (/) or back slashes (\)

NOTE. *Using a back slash as a path separator can produce some unexpected results, depending on how your controller application treats escaped characters. Many applications recognize the sequence of a back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence “\n” might be interpreted as a newline character; “\t” might be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, “C:\\testfile.txt”.*

- Some FILESystem commands can fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed. Refer to the operating system help on file properties for further information.

Table 2-30: File system commands

Command	Description
FILESystem?	Returns the file system state.
FILESystem:COpy	Copies one or more files to a new file.
FILESystem:CWD	Sets or queries the current working directory for FILESystem commands.
FILESystem:DELEte	Deletes a named file or directory.
FILESystem:DIR?	Returns a list of directory contents.
FILESystem:HOMEDir?	Returns the current user's home directory.
FILESystem:LDIR?	Returns a list of directory contents.
FILESystem:MKDir	Makes a new directory.
FILESYSTEM:MOUNT:DRIVE	Mounts a network drive or queries if a network drive is currently mounted.
FILESystem:MOUNT:TEKDrive	Mounts the TekDrive specified by the quoted string arguments.

Table 2-30: File system commands (cont.)

Command	Description
FILESystem:READFile	Copies the named file to the interface.
FILESystem:REName	Assigns a new name to an existing file.
FILESystem:RMDir	Deletes the named directory.
FILESystem:TEKDrive:CODE?	Returns short code in string format.
FILESystem:TEKDrive:CODE:EXPIRYtime?	Returns expiry time of short code.
FILESystem:TEKDrive:CODE:STATUS?	Returns status of short code.
FILESystem:UNMOUNT:DRIVE	Unmount the USB drive.
FILESystem:UNMOUNT:TEKDrive	Unmounts the TekDrive specified by the quoted string argument and the drive name is case insensitive.
FILESystem:WRITEFile	Copies the block data to a named file.

Horizontal command group

Horizontal commands control the time base of the instrument. You can set the time per division (or time per point) of the main time base. You can use the Horizontal commands to do the following:

- Set the scale, horizontal position and reference, and units of the time base
- Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
- Enable or disable the display of the time base

Table 2-31: Horizontal commands

Command	Description
HORizontal?	Queries the current horizontal settings.
HORizontal:ACQDURATION?	Returns the time base duration.
HORizontal:DELAy:MODE	Sets or queries the horizontal delay mode.
HORizontal:DELAy:TIME	Sets or queries the horizontal delay time (position) that is used when delay is on.
HORizontal:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed.
HORizontal:MAIn:INTERPRatio?	Returns the main horizontal time base interpolation ratio.
HORizontal:MODE	Sets or queries the horizontal mode.
HORizontal:MODE:MANual:CONFIgure	Sets or queries which horizontal control (scale or record length) will also change when the sample rate is adjusted.
HORizontal:MODE:RECOrdlength	Sets or queries the record length.
HORizontal:MODE:SAMPLERate	Sets or queries the sample rate.
HORizontal:MODE:SCAle	Sets or queries the horizontal scale.
HORizontal:POSition	Sets or queries the waveform horizontal position, in percent, that is used when delay is off.
HORizontal:PREViewstate?	Returns the display system preview state.
HORizontal:RECOrdlength	Sets or queries the horizontal record length.
HORizontal:SAMPLERate	Sets or queries the horizontal sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide	Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue	Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.
HORizontal:SCAle	Sets or queries the horizontal scale.

Mask command group

Mask commands compare incoming waveforms to standard or user-defined masks. A mask is a set of polygonal regions on the screen. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. Standards with eye patterns usually have three masks, but some have four.

Table 2-32: Mask commands

Command	Description
<code>MASK:DELeTe</code>	Deletes all mask segments of the specified mask test.
<code>MASK:MASK<x>:COUNT?</code>	Returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.
<code>MASK:MASK<x>:COUNT:HITS?</code>	Returns the total number of mask hits in all mask segments for the specified mask test.
<code>MASK:MASK<x>:DEFInedby</code>	Sets or queries whether the specified mask is defined by segments or tolerances.
<code>MASK:MASK<x>:DISPlay</code>	Sets or queries the display state of the mask used for the specified mask test.
<code>MASK:MASK<x>:LIST?</code>	Queries the list of segments in the mask used by the specified mask test.
<code>MASK:MASK<x>:SEG<x>:POINTS</code>	Sets or queries the X/Y coordinates of all points in the designated mask segment.
<code>MASK:MASK<x>:SEG<x>COUNT:HITS?</code>	Returns the total number of mask hits in the specified mask segment of the specified mask test.
<code>MASK:MASK<x>:SOUrce</code>	Sets or queries analog source for the specified mask test.
<code>MASK:MASK<x>:TEST:STATE</code>	Sets or queries state of the specified mask test.
<code>MASK:MASK<x>:TEST:STATUS?</code>	Queries the status of the specified mask test.
<code>MASK:MASK<x>:TEST:THReShold</code>	Sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.
<code>MASK:MASK<x>:TOLerance:HORizontal</code>	Sets or queries the mask horizontal tolerance.
<code>MASK:MASK<x>:TOLerance:UPDatenow</code>	Causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.
<code>MASK:MASK<x>:TOLerance:VERTical</code>	Sets or queries the mask vertical tolerance.
<code>MASK:TEST:WAVEforms</code>	Sets or queries the number of waveform acquisitions to test during mask testing.
<code>RECAll:MASK</code>	Recalls a saved mask definition from a Mask File.

Math command group

Use the commands in the Math Command Group to create and define math waveforms. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.

When a live waveform updates or a reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

NOTE. *Math commands are present once a math has been added.*

Table 2-33: Math commands

Command	Description
DISplay:GLObal:MATH<x>:STATE	Sets or queries the global state of the specified math.
DISplay:SElect:MATH	Sets or queries the overall selected math.
DISplay:WAVEView<x>:MATH:MATH<x>:STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition	Sets or queries the vertical position in divisions of the specified math waveform.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified math waveform.
MATH:ADDNew	Adds the specified math.
MATH:DELeTe	Deletes the specified math.
MATH:LIST?	Lists all currently defined math waveforms.
MATH:MATH<x>:AVG:MODE	Sets or queries the math average mode flag.
MATH:MATH<x>:AVG:WEIGht	Sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging.
MATH:MATH<x>:CAN:SUPPortedfields	Sets or queries the field type for the math for the bus source for CAN bus.

Table 2-33: Math commands (cont.)

Command	Description
MATH:MATH<x>:FUNCTION	Sets or queries the basic math arithmetic function.
MATH:MATH<x>:I2C:SUPPortedfields	Sets or queries the field type for the math for the bus source for I2C bus.
MATH:MATH<x>:INTERpolation	Sets or queries whether sinc interpolation is enabled for math on bus source.
MATH:MATH<x>:LAbel:COLor	Sets or queries color of the specified math's label.
MATH:MATH<x>:LAbel:FONT:BOLD	Sets or queries bold state of the specified math label.
MATH:MATH<x>:LAbel:FONT:ITALic	Sets or queries italic state of the specified math label.
MATH:MATH<x>:LAbel:FONT:SIZE	Sets or queries font size of the specified math label.
MATH:MATH<x>:LAbel:FONT:TYPE	Sets or queries font type of the specified math label.
MATH:MATH<x>:LAbel:FONT:UNDERline	Sets or queries the underline state of the specified math label.
MATH:MATH<x>:LAbel:XPOS	Sets or queries the X screen offset where the math waveform label is displayed.
MATH:MATH<x>:LAbel:YPOS	Sets or queries the Y screen offset where the math waveform label is displayed.
MATH:MATH<x>:DEFine	Defines new waveforms using mathematical expressions.
MATH:MATH<x>:GATing	Specifies or returns the gating setting.
MATH:MATH<x>:LAbel:NAME	Sets or queries the label string.
MATH:MATH<x>:LIN:SUPPortedfields	Sets or queries the field type for the math for the bus source for LIN bus.
MATH:MATH<x>:PARAllel:SUPPortedfields	Sets or queries the field type for the math for the bus source for PARAllel bus.
MATH:MATH<x>:RS232C:SUPPortedfields	Sets or queries the field type for the math for the bus source for RS232C bus.
MATH:MATH<x>:SENT:SUPPortedfields	Sets or queries the field type for the math for the bus source for SENT bus.
MATH:MATH<x>:SIGNeddata	Sets or queries value to denote that bus field is decoded as signed/unsigned data for math on bus source.
MATH:MATH<x>:SOUrce<x>	Sets or queries the specified math source.
MATH:MATH<x>:SPECTral:WINDow	Sets or queries the window function used to multiply the spectral analyzer input data for the specified math waveform.
MATH:MATH<x>:SPI:SUPPortedfields	Sets or queries the field type for the math for the bus source for SPI bus.
MATH:MATH<x>:TYPE	Sets or queries the math type.
MATH:MATH<x>:VUNIT	Sets or queries the math custom vertical units.

Measurement command group

Use the commands in the Measurement Command Group to control the automated measurement system.

Measurement commands can set and query measurement parameters. You can assign parameters, such as waveform sources and reference levels, differently for each measurement.

Edge, gating, and range measurement commands can be either global or per-measurement.

Global edge commands are of the form :MEASUREMENT:XXXX

Global gating commands are of the form :MEASUREMENT:GATING:XXXX

Reference levels for measurements can be global, per-measurement or per-source. The default is global. Per-measurement settings are used when local reference levels are enabled for the measurement with the command :MEASUREMENT:MEAS?:GLOBALREF 0. Per-source settings are used when per-source settings are enabled with the command :MEASUREMENT:REFLEVELS:TYPE PerSource and per-source settings are selected for the measurement with the command :MEASUREMENT:MEAS?:GLOBALREF 0.

Global reference level commands are of the form

```
:MEASUREMENT:REFLEVELS:XXXX
```

Per-source reference level commands are of the form

```
:MEASUREMENT:CH1:REFLEVELS:XXXX
```

```
:MEASUREMENT:MATH1:REFLEVELS:XXXX
```

When there is a change in source type then the measurements should be re-added to get valid results.

```
/* Add a reference slot on which to measure */
:REF:ADDNEW "REF1"
/* Set up the measurement parameters on the reference */
:MEASUREMENT:MEAS1:TYPE AMPLITUDE
:MEASUREMENT:MEAS1:SOURCE REF1
/* Load the new waveform file */
:RECALL:WAVEFORM "E:\waveform.wfm",REF1
*OPC?
/* wait for read from Output Queue. */
/* Take amplitude measurement */
:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
/* change analog to digital waveform file */
:RECALL:WAVEFORM "E:\digital8.wfm",REF1
*OPC?
/* Set up the measurement parameters on the reference
again*/
:MEASUREMENT:MEAS1:TYPE AMPLITUDE
```

```
/* Take amplitude measurement */
:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
```

Table 2-34: Measurement commands

Command	Description
MEASTABLE:ADDNew	Adds a new measurement results view table to the scope application.
MEASTABLE:DELETE	Removes the requested measurement results view table from the scope application.
MEASUREMENT?	This command returns all measurement parameters.
MEASUREMENT:ADDMEAS	This command adds a measurement.
MEASUREMENT:ADDNew	Adds the specified measurement.
MEASUREMENT:ANNOtate	This command sets or queries the annotation state for measurements.
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:HYSTeresis	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:RISEMid	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUREMENT:CH<x>:REFLevels:ABSolute:TYPE	This command sets or queries the reference level type for the source.
MEASUREMENT:CH<x>:REFLevels:BASETop	This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.
MEASUREMENT:CH<x>:REFLevels:METHod	This command sets or queries the method used to calculate reference levels for the source.
MEASUREMENT:CH<x>:REFLevels:PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUREMENT:CH<x>:REFLevels:PERCent:FALLLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:HYSteresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:TYPE	This command specifies or queries the reference level percent type for the source.
MEASUrement:DELeTe	The command deletes the specified measurement.
MEASUrement:DELETEALL	Deletes all the active instances of measurements defined in the scope application.
MEASUrement:EDGE<x>	Sets or queries the type of the edge for the measurement.
MEASUrement:GATing	This command sets or queries the global gating type.
MEASUrement:GATing:ACTive	This command sets or queries the global gating active level used for logic gating.
MEASUrement:GATing:ENDtime	Sets or queries the end gate time for all measurements that use Global gating.
MEASUrement:GATing:HYSteresis	This command sets or queries the global gating hysteresis value used for logic gating.
MEASUrement:GATing:LOGICSource	This command sets or queries the gating data source used for logic gating.
MEASUrement:GATing:MIDRef	This command sets or queries the global gating mid ref value used for logic gating.
MEASUrement:GATing:SEARCHSource	This command sets or queries the global gating search source used for logic gating.
MEASUrement:GATing:STARTtime	Sets or queries the start gate time for all measurements that use Global gating.
MEASUrement:INTERp	This command sets or queries the interpolation mode used to locate edge crossings.
MEASUrement:LIST?	Lists all currently defined measurements.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMId	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE	This command sets or queries the reference level type for the source.
MEASUrement:MATH<x>:REFLevels:BASETop	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:METHod	This command sets or queries the method used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMId	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE	This command specifies or queries the reference level percent type for the source.
MEASUrement:MEAS<x>:BURSTEDGTYPe	This command sets or queries the burst edge type for the measurement.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum?	This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN?	This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum?	This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK?	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPulation?	This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev?	This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum?	This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN?	This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum?	This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK?	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPulation?	This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev?	This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:DELay:EDGE<x>	This command sets or queries the 'to edge' type when EDGE? is EDGE1 and the 'from edge' type when EDGE? is EDG2, for the measurement when the measurement type is DELAY.
MEASUrement:MEAS<x>:DISPlaystat:ENABLE	Turns on and off the display of statistics in measurement badges.
MEASUrement:MEAS<x>:EDGE<x>	This command sets or queries the type of the specified edge, rise or fall, for the measurement.
MEASUrement:MEAS<x>:EDGEIncr	This command sets or queries the edge increment value for the measurement.
MEASUrement:MEAS<x>:EDGES:FROMLevel	This command sets or queries the 'from level' edge for the measurement.
MEASUrement:MEAS<x>:EDGES:LEVel	This sets or queries the level type for the 'time outside level' measurement.
MEASUrement:MEAS<x>:EDGES:N	The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod.
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod	This command sets or queries the slew rate method for the measurement.
MEASUrement:MEAS<x>:EDGES:TOLevel	This command sets or queries the 'to level' edge for the measurement.
MEASUrement:MEAS<x>:FAILCount?	Returns the number of measurement failures, if applicable, for the selected measurement.
MEASUrement:MEAS<x>:FROMedge	This command sets or queries the 'from edge' type for the measurement.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>: FROMEDGESEARCHDIRect	This command sets or queries the 'from edge' search direction for the measurement.
MEASUrement:MEAS<x>:GATing	This command sets or queries the gating type for the measurement.
MEASUrement:MEAS<x>:GATing:ACTive	This command sets or queries the gating active level when the gating type is logic.
MEASUrement:MEAS<x>:GATing:ENDtime	Sets or queries the end gate time for the measurement when using Local gating.
MEASUrement:MEAS<x>:GATing:GLOBal	This command sets or queries the gating settings global flag.
MEASUrement:MEAS<x>:GATing:HYSTeresis	This command sets or queries the gating hysteresis value when the gating type is logic.
MEASUrement:MEAS<x>:GATing: LOGICSource	This command sets or queries the gating data source when the gating type is logic.
MEASUrement:MEAS<x>:GATing:MIDRef	This command sets or queries the gating mid ref value when the gating type is logic.
MEASUrement:MEAS<x>:GATing: SEARCHSource	This command sets or queries the gating search source when the gating type is search.
MEASUrement:MEAS<x>:GATing:STARTtime	Sets or queries the start gate time for the measurement when using Local gating.
MEASUrement:MEAS<x>:GLOBalref	This command sets or queries the reference levels global flag for the measurement.
MEASUrement:MEAS<x>:HIGHREFVoltage	Sets or queries the high reference voltage value for the 'time outside level' measurement.
MEASUrement:MEAS<x>:IDLETime	Sets or queries the idle time for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:LABel	This command sets or queries the label for the measurement.
MEASUrement:MEAS<x>:LOWREFVoltage	This command sets or queries the low reference voltage value for the 'time outside level' measurement.
MEASUrement:MEAS<x>:PASSFAILEnabled	Sets or returns the pass/fail test enable status.
MEASUrement:MEAS<x>:PASSFAILHIGHlimit	Sets or returns the high limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLIMit	Sets or returns the limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLOWlimit	Sets or returns the low limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILMArgin	Sets or returns the allowed margin for limit comparisons for all pass/fail checks
MEASUrement:MEAS<x>:PASSFAILWHEN	Sets or returns the condition on which a measurement test fails.
MEASUrement:MEAS<x>:PERFREQ:EDGE	This command sets or queries the edge type of a Period/Frequency measurement.
MEASUrement:MEAS<x>:POLarity	This command sets or queries the polarity for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:REFLevels: ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:HYSteresis	This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:RISEMid	This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>: ABSolute:TYPE	This command sets or queries the reference level type for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>: BASETop	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>: METHod	This command sets or queries the method used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:FALLLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:HYSteresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:RISEMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>: PERCent:TYPE	This command specifies or queries the reference level percent type for the measurement.
MEASUrement:MEAS<x>:REFMode	This command sets or queries the reference level mode for the measurement.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:REFVoltage	This command sets or queries the reference voltage value for the measurement.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPulation?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum?	This query-only command returns the maximum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN?	This query-only command returns the mean value for the measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum?	This query-only command returns the minimum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK?	This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPulation?	This query-only command returns the population for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?	This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:SIGNALType	Sets or queries the signal type of source 1 for the measurement.
MEASUrement:MEAS<x>:SOUrce<x>	This command sets or queries the measurement source.
MEASUrement:MEAS<x>:STATUS?	Returns the pass fail status, if applicable, for the selected measurement
MEASUrement:MEAS<x>:TOEge	This command sets or queries the 'to edge' type for the measurement.
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect	This command sets or queries the 'to edge' search direction for the measurement.
MEASUrement:MEAS<x>:TRANSition	This command sets or queries the transition edges flag for the measurement.
MEASUrement:MEAS<x>:TYPe	This command sets or queries the measurement type.
MEASUrement:MEAS<x>:XUNIT?	Returns the horizontal scale units of the specified measurement.
MEASUrement:MEAS<x>:YUNIT?	Returns the vertical scale units of the specified measurement.
MEASUrement:REFLevels:ABSolute:FALLHigh	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:FALLLow	Sets or queries the value used as the low reference level of the falling edge.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:REFLevels:ABSolute:FALLMid	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:HYSTeresis	Sets or queries the value of the hysteresis of the reference level.
MEASUrement:REFLevels:ABSolute:RISEHigh	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISELow	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISEMid	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:TYPE	Sets or queries the reference level type.
MEASUrement:REFLevels:BASETop	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REFLevels:METHod	Sets or queries the method used to calculate reference levels.
MEASUrement:REFLevels:MODE	This command sets or queries how often reference levels are calculated.
MEASUrement:REFLevels:PERCent:FALLHigh	Sets or queries the percentage used to calculate the high reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLLow	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLMid	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:HYSTeresis	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REFLevels:PERCent:RISEHigh	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISELow	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISEMid	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REFLevels:PERCent:TYPE	Sets or queries the reference level percent type.
MEASUrement:REFLevels:TYPE	This command sets or queries the shared reference level method used for sources of measurement calculations.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow	Sets or queries the value used as the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis	Sets or queries the value of the hysteresis of the reference level.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE	Sets or queries the reference level type.
MEASUrement:REF<x>:REFLevels:BASETop	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:METHod	Sets or queries the method used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh	Sets or queries the percentage used to calculate the high reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow	Sets or queries the percentage used to calculate the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:HYSteresis	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISELow	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:TYPE	Sets or queries the reference level percent type.
MEASUrement:STATIstics:CYCLEMode	This command sets or queries whether cycle-cycle statistics are calculated for all measurements.

Miscellaneous command group

Miscellaneous commands do not fit into other categories.

Several commands and queries are common to all devices. The 488.2-1987 standard defines these commands. The common commands begin with an asterisk (*) character.

Table 2-35: Miscellaneous commands

Command	Description
AUTOSAVEPITIMEOUT	Sets or queries the idle time from the programmable interface before auto-save occurs.
AUTOSAVEUITIMEOUT	Sets or queries the idle time from the user interface before auto-save occurs.
AUTOSet	Sets or queries the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.
AUTOSet:ACQuisition:ENable	Sets or queries the Autoset acquisition setting adjustment.
AUTOSet:ENable	Sets or queries the Autoset enable/disable mode.
AUTOSet:HORizontal:ENable	Sets or queries Autoset's adjustment of horizontal settings.
AUTOSet:TRIGger:ENable	Sets or queries Autoset's adjustment of trigger settings.
AUTOSet:VERTical:ENable	sets or queries Autoset's adjustment of vertical settings.
AUTOSet:VERTical:OPTIMize	Sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode.
AUXout:EDGE	Sets or queries the direction in which the trigger output signal will transition when a trigger occurs.
AUXout:SOUrce	Sets or queries the trigger source at the BNC connection.
CLEAR	Clears acquisitions, measurements, and waveforms.
CONNected:REQUested:STATus	Sets the status of the Connected Scope Preference feature.
CONNected:STATus?	Returns the status of the Connected Scope Preference feature.
CONNected:USAGe:TRack:REQUested:STATus	Sets the tracking usage status of the Connected Scope Preference feature.
CONNected:USAGe:TRack:STATus?	Returns the tracking usage status of the Connected Scope Preference feature.
DATE?	Queries the date that the instrument displays.
*DDT	Sets or queries the commands that will be executed by the group execute trigger.
FPAnel:PRESS	Emulates a button press.
FPAnel:TURN	Emulates a knob turn.
HEADer	Sets or queries the Response Header Enable State.
ID?	Returns identifying information about the instrument and its firmware.
*IDN?	Returns the instrument identification code.
LICense?	Queries all license parameters.
LICENSE:APPID?	Returns a comma-separated list of the active application IDs.
LICense:COUNt?	Returns a count of the number of active licenses installed.

Table 2-35: Miscellaneous commands (cont.)

Command	Description
LICense:ERRor?	This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue (alias for ALLEV?).
LICense:GMT?	Returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.
LICense:HID?	Returns the instrument HostID unique identifier.
LICense:INSTall	Accepts a <block data> license and installs it on the instrument.
LICense:ITEM?	Returns the details pertaining to a specific license.
LICense:LIST?	Returns the active license nomenclatures as a comma-separated list of strings.
LICense:VALidate?	Accepts a license nomenclature as an argument and returns the status of the license.
LIC:UNINSTALL?	Returns the exit license information for the user to return to their TekAMS account.
LOCK	Sets or queries the front panel and touchscreen lock state.
*LRN?	Returns a listing of instrument settings.
MAINWindow:BADGe:BRINGtoview	Makes a specified badge visible.
MAINWindow:FONTSize	Sets the font size for UI text elements.
MAINWindow:RRBDisplaystate	Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).
NEWpass	Changes the password for user protected data.
PASSWord	Provides access for changing user protected data.
PAUSe	Causes the interface to pause the specified number of seconds before processing any other commands.
REM	Specifies a comment which is ignored by the instrument.
ROSc:SOUrce	Selects or queries the selected source for the time base reference oscillator.
ROSc:STATE?	Returns whether the time base reference oscillator is locked.
SCOPEApp REBOOT	Reboots the scope.
SET?	Returns a listing of instrument settings.
SOCKETServer:ENable	Enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument.
SOCKETServer:PORT	Sets the TCPIP port for the socket server connection.
SOCKETServer:PROTOCol	Sets or queries the protocol for the socket server.
TEKSecure	Initializes both waveform and setup memories.
TIMe	Queries the time displayed by the instrument.
TIMe:ZONe	Sets the time zone to the one specified.
TIMe:ZONe:UTCDELTA	Sets or queries the time zone using the difference between the desired time zone and UTC.
TOTALuptime?	Returns the total number of hours the instrument has been turned on since the NV memory was last programmed.
TOUCHSCReen:STATe	Sets or queries the enabled state of the touch screen.

Table 2-35: Miscellaneous commands (cont.)

Command	Description
*TRG	Performs the group execute trigger (GET).
*TST?	Tests the interface and returns status.
UNDO	Reverts the scope settings to a state before the previous command or user interface action.
UNLock	Unlocks front panel controls.
USBDevice:CONFigure	Used to configure the rear USB port to be off or enabled as a USBTMC device.
VERBose	Sets or queries the verbose state.

Pattern Generator group

Configure the Pattern Generator to provide output of a predefined pattern up to bit rate 25 Mb/s.

Table 2-36: Pattern Generator commands

Command	Description
PG:AMPlitude	Sets or queries the Pattern Generator output voltage.
PG:BIT:ONE	Sets or queries the output value of Pattern Generator bit 1.
PG:BIT:THREE	Sets or queries the output value of Pattern Generator bit 3.
PG:BIT:TWO	Sets or queries the output value of Pattern Generator bit 2.
PG:BIT:ZERO	Sets or queries the output value of Pattern Generator bit 0.
PG:BITRate	Sets or queries the bit rate of data in the Pattern Generator.
PG:BURSt:CCOUnt	Sets or queries the cycle count for Pattern Generator burst mode.
PG:BURSt:TRIGger	Manually starts the burst pattern.
PG:FILE:PATtern	Sets or queries the path of your data file to generate a digital pattern in file mode.
PG:OUTPut:MODe	Sets or queries the Pattern Generator output mode.
PG:PATTERNdefinition	Sets or queries the Pattern Generator definition.

Plot command group

Plot commands let you select the type and control the appearance of your plots.

Table 2-37: Plot commands

Command	Description
PLOT:ADDNew	Adds the specified plot.
PLOT:DELeTe	Deletes the specified plot.
PLOT:LIST?	Lists all currently defined plots.
PLOT:PLOT<x>:SOUrce<x>	Sets or queries the measurement source.
PLOT:PLOT<x>:TYPe	Sets or queries the current plot type for the selected plot.

Save and Recall command group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

Table 2-38: Save and Recall commands

Command	Description
FACTory	Resets the instrument to factory default settings.
RECAII:SESSion	Restores the state of the instrument from a saved session file.
RECAII:SETUp	Recalls saved instrument settings.
RECAII:WAVEform	Recalls a stored waveform to a reference memory location.
SAVe:EVENTtable:BUS	Saves bus results table to the specified file.
SAVe:EVENTtable:MEASUrement	Saves data (measurement) results to the specified file.
SAVe:EVENTtable:SEARCHTable	Saves a search results table to the specified file.
SAVe:IMAGe	Saves a capture of the screen contents to the specified image file.
SAVe:IMAGe:COMPosition	Sets or queries the color mode for saved images (normal or inverted).
SAVe:IMAGe:VIEWType	Sets or queries the view type for saved images. Currently only FULLScreen is supported.
SAVe:PLOTData	Saves the plot data of the currently selected plot in a specified file. Supported file format is CSV.
SAVe:REPOrt	Saves a report to the specified file or, if no argument is specified, uses the folder and file name specified by the related commands.
SAVe:REPOrt:COMMENTS	Sets or queries the comments to be included in saved report files.
SAVe:SESSion	Saves the state of the instrument, including reference waveforms, to a saved session file.
SAVe:SETUp	Saves the current instrument state to the specified file.
SAVe:SETUp:INCLUDEREFs	Sets or queries whether displayed reference waveforms are to be included in saved setups.
SAVe:WAVEform	Saves the specified waveform(s) to the specified destination file(s).
SAVe:WAVEform:GATing	Sets the method to save a specified part of the waveform data.
SAVe:WAVEform:GATing:RESAMPLErate	Saves the waveform data at a sample interval.
SAVe:WAVEform:SOURCEList?	Returns a list of the available waveforms that can be specified as the source for the SAVe:WAVEform command.

Save On command Group

Use this group of commands to program the instrument to save images, measurements, waveforms, or the instrument setup, on triggers that you select. These commands still function, however the **Act On Event** commands are preferred. Please see the Act On Event section for continued development and enhancements. (See Table 2-18.)

Table 2-39: Save On commands

Command	Description
SAVEON:FILE:DEST	Sets or queries the location where files are saved.
SAVEON:FILE:NAME	Sets or queries the file name to use when SAVEON:TRIGger is ON.
SAVEON:IMAGe:FILEFormat	Sets or queries the file format to be used for saved image files.
SAVEON:IMAGe	Sets or queries whether to save a screen capture when a trigger occurs.
SAVEON:TRIGger	Sets or queries whether to save a file when a trigger occurs.
SAVEON:WAVEform	Sets or queries whether to save a waveform when a limit test failure, mask failure, or trigger occurs.
SAVEON:WAVEform:FILEFormat	Sets or queries the file format for saving waveform.
SAVEON:WAVEform:SOURce	Sets or queries the sources for saving waveforms.

Search and Mark command group

Use search and mark commands to seek out and identify information in waveform records that warrant further investigation.

Table 2-40: Search and Mark commands

Command	Description
SEARCH:ADDNew	Adds the specified search.
SEARCH:DELete	Deletes the specified search.
SEARCH:DELETEALL	Deletes all the active instances of measurements defined in the scope application.
SEARCH:LIST?	Lists all currently defined searches.
SEARCH:SEARCH<x>:COPY	Copies the search criteria to or from the trigger.
SEARCH:SEARCH<x>:NAVigate	Sets the navigation action for search marks.
SEARCH:SEARCH<x>:TOTAL?	Queries the total number of found search marks for this search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition	Sets or queries the CAN bus trigger condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:DIRection	Sets the CAN search type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:OFFSet	Sets or queries the data offset value, in bytes, to use when searching on the CAN data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:QUALifier	Sets or queries the CAN bus trigger data qualifier.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:SIZE	Sets or queries the CAN bus trigger data size.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:VALue	Sets or queries the binary data value to be used when searching on a CAN bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType	Sets or queries the type of error condition for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit	Sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit	Sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEType	Sets or queries CAN bus trigger frame type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODE	Sets or queries the CAN bus trigger identifier mode.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue	Sets or queries CAN bus trigger identifier value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODE	Sets or queries the address mode for the specified I2C bus trigger search to determine where to place a mark.

Table 2-40: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue	Sets or queries the address string when the search condition for the specified search is Address or AddressData, to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition	Sets or queries the trigger condition for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection	Sets or queries the direction of the data for the I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZE	Sets or queries the length of the data string in bytes used for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue	Sets or queries the data value of the data token for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue	Specifies the high data value to be used in a LIN search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier	Sets or queries the LIN bus search trigger data qualifier.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZE	Sets or queries the LIN bus search trigger data size.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue	Sets or queries the LIN bus search trigger data value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE	Sets or queries the LIN bus search trigger error type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue	Sets or queries the LIN bus search trigger identifier value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARAllel:DATa:VALue	Sets or queries the data value for a parallel bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue	Sets or queries the data string value for the specified RS232c bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition	Sets or queries the condition for an RS232C bus search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZE	Sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition	Sets or queries the search condition for a SENT bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType	Sets or queries the error type to be used when searching on SENT data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC	Sets or queries the CRC error type to be used when searching on SENT data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue	Sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.

Table 2-40: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue	Sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue	Sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue	Sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:HIVALue	Sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:VALue	Sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue	Sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATUS:VALue	Sets or queries the binary status value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier	Sets or queries the qualifier to be used when searching on SENT pause pulses.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue	Sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue	Sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue	Sets or queries the high binary Slow channel data value to use when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier	Sets or queries the qualifier to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue	Sets or queries the binary slow channel data value to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue	Sets or queries the binary Slow identifier value to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOURce	Sets or queries the bus source for the specified bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition	Sets or queries the trigger condition for the specified SPI bus trigger search to determine where to place a mark.

Table 2-40: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZE	Sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue	Sets or queries the data value of the data token for the specified SPI bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCEType	Sets or queries trigger Source for SPI bus.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe	Sets or queries the slope for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce	Sets or queries the source waveform for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion	Sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge	Sets whether or not Logic search uses a clock source.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn	Sets or queries the condition setting for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth	Sets or queries the width setting for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:EDGE	Sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOURce	Sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime	Sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime	Sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME	Sets or queries the time setting for a timeout trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TYPe	Sets or queries the trigger type setting for a search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition	Sets or queries the LIN bus search trigger condition.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold	Sets or queries the source threshold level for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCK:THReshold	Sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTime	Specifies the Logic search delta time value.

Table 2-40: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPUT:CLOCK:SOUrce	Sets or queries the channel to use as the clock source for logic trigger.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVel:CH<x>	Sets or queries the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVel:MATH<x>	Sets the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVel:REF<x>	Sets the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:CH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:DCH<x>_D<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:MATH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:REF<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:POLarity	Sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type.
SEARCH:SEARCH<x>:TRIGger:A:LOGic:WHEn	Sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit	Specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification	Specifies whether or not to use logic qualification for a pulse width search.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit	Specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity	Specifies the polarity for a pulse width search.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce	Sets and queries the source for the pulsewidth search input.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold	Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn	Specifies to search for a pulse with a specified width.

Table 2-40: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification	Specifies whether or not to use logic qualification for a runt search.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity	Specifies the polarity for the runt search.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce	Sets and queries the source for the Runt search input.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH	Sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW	Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold	Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>	Sets and returns the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:STATE	Sets or queries the enabled state of the search.
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq	Sets or queries whether acquisitions are stopped when a search hit is found.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification	Sets whether or not to use logic qualification for a timeout search.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity	Sets or queries the polarity to be used for a Timeout search.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce	Sets and queries the source for timeout search input.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold	Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark.
SEARCH:SELected	Sets or queries the selected search.

Self Test command group

The Self Test commands control the selection and execution of diagnostic tests.

Table 2-41: Self Test commands

Command	Description
DIAG:LOOP:OPTion	Sets or queries the type of looping desired.
DIAG:LOOP:OPTion:NTIMes	Sets or queries how many loops to run.
DIAG:LOOP:STOP	Stops diagnostics looping.
DIAG:MODe	Sets or queries the diagnostics mode.
DIAG:RESUlt?	Returns the diagnostics results.
DIAG:RESUlt:FLAg?	Returns the status of the diagnostics (single area).
DIAG:RESUlt:LOG?	Returns the status of the diagnostic area.
DIAG:SElect	Selects or queries an available diagnostic area.
DIAG:STATE	Sets the instrument operating state.

Status and Error command group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (*) character.

Table 2-42: Status and Error commands

Command	Description
ALLEV?	Returns all events and their messages.
BUSY?	Returns instrument status.
*CLS	Clears status.
DESE	Sets or queries the bits in the Device Event Status Enable Register.
*ESE	Sets or queries the bits in the Event Status Enable Register.
*ESR?	Returns the contents of the Standard Event Status Register.
EVENT?	Returns event code from the event queue.
EVMsg?	Returns event code, message from the event queue.
EVQty?	Returns number of events that are enabled in the queue.
*OPC	Generates the operation complete message in the standard event status register when all pending operations are finished Or returns "1" when all current operations are finished.
*OPT?	Returns a comma separated list of installed options as an arbitrary ASCII string (no quotes).
*PSC	Sets or queries the power on status flag.
*PUD	Sets or queries a string of protected user data.
*RST	Resets the instrument to factory default settings.
*SRE	Sets or queries the bits in the Service Request Enable Register.
*STB?	Returns the contents of the Status Byte Register.
*WAI	Prevents the instrument from executing further commands until all pending operations finish.

Trigger command group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There is one trigger: A. You can set the A triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set A triggers to Pulse Width, Timeout, Runt, Logic, Setup & Hold, Rise/Fall Time, and Bus types.

Table 2-43: Trigger commands

Command	Description
TRIGger	Forces a trigger event to occur or returns current trigger parameters for the instrument.
TRIGger:A:BUS:B<x>:CAN:CONDition	Sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:DATA:DIRection	Sets the data direction (read, write or “nocare”) to be used to search on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:DATA:OFFSet	Sets or queries the data offset value, in bytes, to use when triggering on the CAN data field.
TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier	Sets the qualifier (<, >, =, not =, <=, >=) to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:DATA:SIZE	Sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:DATA:VALue	Sets the binary data value to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:ERRType	Sets or queries the type of error condition for a CAN bus to triggering on.
TRIGger:A:BUS:B<x>:CAN:FD:BRSBit	Sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on.
TRIGger:A:BUS:B<x>:CAN:FD:ESIBit	Sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on.
TRIGger:A:BUS:B<x>:CAN:FRAMeType	Sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:IDentifier:MODE	Sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:CAN:IDentifier:VALue	Sets the binary address value to be used when triggering on a CAN bus signal.
TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE	Specifies the I ² C address mode to 7 or 10-bit.
TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue	Specifies the binary address string used for the I ² C trigger if the trigger condition is ADDRESS or ADDRANDDATA.
TRIGger:A:BUS:B<x>:I2C:CONDition	Specifies the trigger condition for an I ² C trigger.
TRIGger:A:BUS:B<x>:I2C:DATA:DIRection	Specifies the I ² C trigger type to be valid on a Read, Write, or No Care condition.
TRIGger:A:BUS:B<x>:I2C:DATA:SIZE	Specifies the length of the data string in bytes to be used for an I ² C trigger if the trigger condition is DATA or ADDRANDDATA.

Table 2-43: Trigger commands (cont.)

Command	Description
TRIGger:A:BUS:B<x>:I2C:DATA:VALue	Specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA.
TRIGger:A:BUS:B<x>:LIN:CONDition	Specifies the trigger condition for LIN.
TRIGger:A:BUS:B<x>:LIN:DATA:HIVALue	Specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OTRANGE.
TRIGger:A:BUS:B<x>:LIN:DATA:QUALifier	Specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.
TRIGger:A:BUS:B<x>:LIN:DATA:SIZE	Specifies the length of the data string in bytes to be used for LIN trigger.
TRIGger:A:BUS:B<x>:LIN:DATA:VALue	Specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA.
TRIGger:A:BUS:B<x>:LIN:ERRType	Specifies the error type be used for LIN trigger.
TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue	Specifies the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.
TRIGger:A:BUS:B<x>:PARAllel:DATA:VALue	Specifies the binary data string used for a Parallel Bus trigger.
TRIGger:A:BUS:B<x>:RS232C:CONDition	Specifies the condition for an RS-232C trigger.
TRIGger:A:BUS:B<x>:RS232C:DATA:SIZE	Sets or queries the length of the data string in bytes to be used for an RS232 trigger.
TRIGger:A:BUS:B<x>:RS232C:DATA:VALue	Sets or queries the data value of the data token for an RS232 trigger.
TRIGger:A:BUS:B<x>:SENT:CONDition	Sets or queries the trigger condition for a SENT bus.
TRIGger:A:BUS:B<x>:SENT:ERRType	Sets or queries the error type to be used when triggering on SENT data.
TRIGger:A:BUS:B<x>:SENT:ERRType:CRC	Sets or queries the CRC error type to be used when triggering on SENT data.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue	Sets or queries the high binary Fast channel 1 value to use when triggering on a SENT bus signal.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier	Sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 1.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue	Sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue	Sets or queries the high binary fast channel 2 value to use when triggering on a SENT Fast bus signal.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier	Sets or queries the qualifier to be used when triggering on SENT Fast packet bus data for device channel 2.
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:VALue	Sets or queries the binary Fast channel 2 value to be used when triggering on a SENT bus signal. The trigger condition must be set to FAST.
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:HIVALue	Sets or queries the high binary Fast message counter value to be used when triggering on a SENT bus signal.
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:QUALifier	Sets or queries the qualifier to be used when triggering on SENT Fast packet bus data for the secure format counter.
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:VALue	Sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.
TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue	Sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.

Table 2-43: Trigger commands (cont.)

Command	Description
TRIGger:A:BUS:B<x>:SENT:FAST:STATus:VALue	Sets or queries the binary status value to be used when triggering on a SENT bus signal.
TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier	Sets or queries the qualifier to be used when triggering on SENT pause pulses.
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:HIVALue	Sets or queries the high binary Slow channel data value to use when triggering on SENT Slow packet bus data.
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:QUALifier	This command sets or queries the binary identifier value to use when triggering on Slow packet bus data.
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:VALue	This command sets or queries the binary slow channel value to use when triggering on Slow packet bus data.
TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:VALue	This command sets or queries the qualifier to use when triggering on SENT Slow packet bus data.
TRIGger:A:BUS:B<x>:SPI:CONDition	Specifies the trigger condition for a SPI trigger.
TRIGger:A:BUS:B<x>:SPI:DATA:SIZE	Specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATA.
TRIGger:A:BUS:B<x>:SPI:DATA:VALue	Specifies the binary data string used for SPI triggering if the trigger condition is DATA.
TRIGger:A:BUS:SOUrce	Sets or queries the source for a bus trigger.
TRIGger:A:EDGE:COUPling	Sets or queries the type of coupling for the edge trigger.
TRIGger:A:EDGE:SLOpe	Sets or queries the slope for the edge trigger.
TRIGger:A:EDGE:SOUrce	Sets or queries the source for the edge trigger.
TRIGger:A:LEVel:CH<x>	Sets or queries the level for the trigger for the channel.
TRIGger:A:LOGic:DELTAtime	Sets or queries the Logic trigger delta time value.
TRIGger:A:LOGic:FUNcTION	Sets or queries the logical combination of the input channels for the logic trigger.
TRIGger:A:LOGic:INPut:CLOCK:SOUrce	Sets or queries the channel to use as the clock source.
TRIGger:A:LOGic:POLarity	Sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.
TRIGger:A:LOGic:USECLockedge	Sets whether or not Logic trigger type uses clock source.
TRIGger:A:LOGic:WHEN	Sets or queries the condition for generating a logic trigger with respect to the defined input pattern.
TRIGger:A:LOGICPattern:CH<x>	Sets or queries the Logic Pattern to trigger on.
TRIGger:A:LOGICPattern:DCH<x>_D<x>	Sets or queries the Logic Pattern to trigger on.
TRIGger:A:LOWerthreshold:CH<x>	Sets or queries CH<x> lower trigger level.
TRIGger:A:PULSEWidth:HIGHLimit	Sets or queries the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:A:PULSEWidth:LOWLimit	Sets or queries the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:A:PULSEWidth:POLarity	Sets or queries the polarity for a pulse width trigger.
TRIGger:A:PULSEWidth:SOUrce	Sets or queries the source waveform for a pulse width trigger.
TRIGger:A:PULSEWidth:WHEN	Sets or queries to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value.

Table 2-43: Trigger commands (cont.)

Command	Description
TRIGger:A:RUNT:POLarity	Sets or queries the polarity for the runt trigger.
TRIGger:A:RUNT:SOUrce	Sets or queries the source waveform for the runt trigger.
TRIGger:A:RUNT:WHEn	Sets or queries the type of pulse width the trigger checks for when it detects a runt.
TRIGger:A:RUNT:WIDTh	Sets or queries the width, in seconds, for a runt trigger.
TRIGger:A:SETHold:CLOCK:EDGE	Sets or queries the clock edge polarity for setup and hold triggering.
TRIGger:A:SETHold:CLOCK:SOUrce	Sets or queries the clock source for the setup and hold triggering.
TRIGger:A:SETHold:HOLDTime	Sets or queries the hold time for setup and hold violation triggering.
TRIGger:A:SETHold:SETTime	Sets or queries the setup time for setup and hold violation triggering.
TRIGGER:A:SETHOLDLOGICVAL: DCH<x>_D<x>	Sets or queries the conditions used for generating an A logic pattern.
TRIGger:A:TIMEOut:POLarity	When triggering using the <code>TIMEOUT</code> trigger type, this commands specifies the polarity to be used.
TRIGger:A:TIMEOut:SOUrce	When triggering using the <code>TIMEOUT</code> trigger type, this command specifies the source.
TRIGger:A:TIMEOut:TIME	When triggering using the <code>TIMEOUT</code> trigger type, this command specifies the timeout time, in seconds.
TRIGger:A:TYPe	Sets or queries the type of A trigger.
TRIGger:A:UPPPerthreshold:CH<x>	Sets or queries the CH<x> upper trigger level.
TRIGger:A	Sets the A trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal.
TRIGger:A:HOLDOff:TIME	Sets or queries the A trigger holdoff time.
TRIGger:A:MODE	Sets or queries the A trigger mode.
TRIGger:AUXLevel	Sets or queries the Auxiliary Input voltage level to use for an edge trigger.
TRIGger:HYSTeresis:USER:STATe	Enables or disables user-defined trigger hysteresis.
TRIGger:HYSTeresis:USER:VALue	Sets or returns the height of the user-defined trigger hysteresis zone in volts.
TRIGger:STATE?	Queries the current state of the triggering system.

Vertical command group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms.

You can replace VOLts with SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-44: Vertical commands

Command	Description
CH<x>?	Returns vertical parameters for the specified channel.
CH<x>:BANdwidth	Sets or queries the bandwidth of the specified channel.
CH<x>:CLIPping?	Queries whether the specified channel's input signal is clipping (exceeding) the channel vertical scale setting.
CH<x>:COUPling	Sets or queries the coupling setting for the specified channel.
CH<x>:DESKew	Sets or queries the deskew time for the specified channel.
CH<x>:DITHErRange	Sets or queries the amount of dithering for the specified analog channel.
CH<x>:INVert	Sets or queries invert state of the specified channel.
CH<x>:LABel:COLor	Sets or queries the color of the specified channel label.
CH<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified channel label.
CH<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified channel label.
CH<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified channel label.
CH<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified channel label.
CH<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified channel label.
CH<x>:LABel:NAME	Defines or queries the label for the channel waveform.
CH<x>:LABel:XPOS	Sets or queries the X display coordinate for the channel waveform label.
CH<x>:LABel:YPOS	Sets or queries the Y display coordinate for the channel waveform label.
CH<x>:OFFSet	Sets or queries the channel offset.
CH<x>:POSition	Sets or queries the vertical position for the specified analog channel.
CH<x>:PROBEFunc:EXTAtten	Sets the attenuation value for the specified channel to the specified scale factor. Or queries the user-specified attenuation.
CH<x>:PROBEFunc:EXTDBatten	Sets the attenuation value for the specified channel to the specified value in decibels. Or queries the user-specified attenuation in decibels.
CH<x>:PROBEFunc:EXTUnits	Sets or queries the alternate unit for the external attenuator of the specified channel.
CH<x>:PROBEFunc:EXTUnits:STATE	Sets or queries the custom units enable state for the specified channel.
CH<x>:SCAle	Sets or returns the vertical scale for the specified analog channel.
CH<x>:SCALERATio	Sets or returns the scale ratio for the specified analog channel.
CH<x>:TERmination	Sets channel input termination.

Table 2-44: Vertical commands (cont.)

Command	Description
CH<x>:VTERm:BIAS	Sets or queries the termination voltage for the specified channel (if control is available).
CONFIguration:ANALOG:BANDWidth?	Queries the maximum licensed bandwidth of the instrument.
REF:ADDNew	Adds the specified reference. Argument is of the form "REF<NR1> ", where $NR1 \geq 1$.
REF:DELeTe	Deletes the specified reference.
REF:REF<x>:DESKew	Sets or queries the deskew time for the specified reference.
REF:REF<x>:LABel:COLor	Sets or queries the color of the specified ref label.
REF:REF<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified reference label.
REF:REF<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified reference label.
REF:REF<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified reference label.
REF:REF<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified reference label.
REF:REF<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified reference label.
REF:REF<x>:LABel:NAME	Sets or queries the label of the designated waveform.
REF:REF<x>:LABel:XPOS	Sets or queries the position of the reference waveform label on the X axis.
REF:REF<x>:LABel:YPOS	Sets or queries the position of the reference waveform label on the Y axis.
REF:LIST?	Lists all currently defined references.
REF:REF<x>:SOUrce	Sets or queries the filename used by the given reference.
VERTical:DESKew:FROM:CuSTOMPROPAGation	Sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.
VERTical:DESKew:STATIC	Sets or queries the target channel for performing channel-to-channel deskew adjustment.
VERTical:DESKew:TO:CuSTOMPROPAGation	Sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.
VERTical:DESKew:FROM:SOUrce	Sets or queries the source channel for performing channel-to-channel deskew adjustment.
VERTical:DESKew:TO:SOUrce	Sets or queries the target channel for performing channel-to-channel deskew adjustment.

Waveform Transfer command group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.

Data formats. Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple or ENVELOpe mode uses eight bits per waveform data point. Data acquired in AVERage mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the DATA:ENCdg command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

ASCII data is represented by signed integer or floating point values. An example ASCII waveform data string can look like this:

```
CURVE<space>-110,-109,-110,-110,-109,-107,  
-109,-107,-106,-105,-103,-100,-97,-90,-84,-80
```

Use ASCII to obtain more readable and easier to format output than binary. However, ASCII can require more bytes to send the same values than it does with binary. This can reduce transmission speeds.

Binary data can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a MATH (or REF that came from a MATH) is used, 32-bit floating point values are used that are four bytes in width.

The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- RIBinary specifies signed integer data-point representation with the most significant byte transferred first.
- SRIBinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- RFBinary specifies floating point data-point representation with the most significant byte transferred first.
- SRFBinary is the same as RFBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

Waveform data and record lengths. You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the `DATA:START` and `DATA:STOP` commands to specify the first and last data points of the waveform record.

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting `DATA:START` to 1 and `DATA:STOP` to the record length will always return the entire waveform.

Waveform data locations and memory allocation. The `DATA:SOURce` command specifies the waveform source when transferring a waveform from the instrument.

Waveform preamble. Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual `WFMOupre?` commands for more information.

Scaling waveform data. Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble.

Transferring waveform data from the instrument.

You can transfer waveforms from the instrument to an external controller using the following sequence:

1. Select the waveform source(s) using `DATA:SOURce`.
2. Specify the waveform data format using `DATA:ENCdg`.
3. Specify the number of bytes per data point using `WFMOupre:BYT_Nr`.

NOTE. *MATH waveforms (and REF waveforms that came from a MATH) are always set to four bytes.*

4. Specify the portion of the waveform that you want to transfer using `DATA:START` and `DATA:STOP`.
5. Transfer waveform preamble information using `WFMOupre`.
6. Transfer waveform data from the instrument using `CURVe?`.

Table 2-45: Waveform Transfer commands

Command	Description
<code>CURVe?</code>	The query format transfers waveform data from instrument specified by the <code>DATA:SOURce</code> command.
<code>CURVStream?</code>	Continuously transfers waveform data from the instrument as it is acquired.
<code>DATA</code>	Sets the format and location of the waveform data that is transferred with the <code>CURVe</code> Command. Or queries the format and location of the waveform data that is transferred with the <code>CURVe?</code> command.
<code>DATA:ENCdg</code>	Sets or queries the format of outgoing waveform data.
<code>DATA:MODE</code>	Sets or returns the mode for waveform data.
<code>DATA:RESample</code>	Sets or queries the resampling of outgoing waveform data.
<code>DATA:SOURce</code>	Sets or queries the location of waveform data transferred from the instrument.
<code>DATA:SOURce:AVAILable?</code>	Returns a list of enumerations representing the source waveforms that are currently available for <code>:CURVe?</code> queries.
<code>DATA:START</code>	Sets or queries the starting point in waveform transfer.
<code>DATA:STOP</code>	Sets or queries the ending data point in waveform transfer.
<code>DATA:WIDTH</code>	Specifies the width, in bytes per point, for waveform data transferred from the instrument via the <code>CURVe?</code> query.
<code>WAVFrm?</code>	Returns <code>WFMOupre?</code> and <code>CURVe?</code> data for the waveform as specified by the <code>DATA:SOURce</code> command.
<code>WFMOupre?</code>	Returns the waveform formatting data for the waveform specified by the <code>DATA:SOURce</code> command.
<code>WFMOupre:ASC_Fmt?</code>	Returns the format for ASCII data transferred from the instrument.

Table 2-45: Waveform Transfer commands (cont.)

Command	Description
WFMOupre:BIT_Nr	Sets or queries the number of bits per waveform point that outgoing waveforms contain.
WFMOupre:BN_Fmt	Sets or queries the format of binary data for the waveform.
WFMOupre:BYT_Nr	Sets or queries the data width for the waveform.
WFMOupre:BYT_Or	Sets or queries the byte order of waveform points.
WFMOupre:CENTERFREQuency?	Returns the center frequency of the outgoing waveform.
WFMOupre:DOMain?	Returns the domain of the outgoing waveform.
WFMOupre:ENCdg	Sets or queries the type of encoding for outgoing waveforms.
WFMOupre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query.
WFMOupre:PT_Fmt?	Returns the point format for the waveform.
WFMOupre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform.
WFMOupre:PT_ORder?	Returns whether the source waveform is Fast Acquisition.
WFMOupre:RESample	Sets or queries the resampling of outgoing waveform data.
WFMOupre:SPAN?	Returns the frequency span of the outgoing waveform.
WFMOupre:WFId?	Returns a string describing the acquisition parameters for the waveform.
WFMOupre:WFMTYPE?	Returns the type of the outgoing waveform.
WFMOupre:XINcr?	Returns the horizontal sampling interval.
WFMOupre:XUNit?	Returns the horizontal units for the waveform.
WFMOupre:XZEr0?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger.
WFMOupre:YOFf?	Returns the vertical offset in digitizing levels for the waveform.
WFMOupre:YUNit?	Returns the vertical units for the waveform.
WFMOupre:YZEr0?	Returns the vertical offset for the waveform.

Zoom command group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

NOTE. *Zoom commands are available once a view has been added.*

Table 2-46: Zoom commands

Command	Description
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM	Sets or queries the start of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO	Sets or queries the end of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM	Sets or queries the start of the zoom y axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO	Sets or queries the end of the zoom y axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM	Sets or queries the start of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:TO	Sets or queries the end of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM	Sets or queries the start of the zoom y-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:TO	Sets or queries the end of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM	Sets or returns the left edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO	Sets or queries the right edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM	Sets or queries the bottom value of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO	Sets or queries the top value of the zoom y-axis in the specified plot view.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition	Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALE	Sets or queries the zoom horizontal scale factor, of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALE	Sets or queries the zoom window horizontal scale in the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe	Sets or queries the specified zoom in the specified Waveform View on or off.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition	Sets or queries the vertical position of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE	Sets or queries the zoom vertical scale factor of the specified waveform for the specified zoom in the specified Waveform View.
DISplay:WAVEView<x>:Zoom?	Queries the zoom parameters of the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>?	Queries the zoom parameters of the specified zoom in the specified Waveform View.

Commands listed in alphabetical order

NOTE. Some of the following commands may not be available on your instrument model. Also, some of the following commands are only available if your instrument has the associated option installed.

ACQUIRE? (Query Only)

Queries the current acquisition state.

Group Acquisition

Syntax ACQUIRE?

Examples ACQUIRE? might return :ACQUIRE:STATE 1;NUMENV INFINITE;STOPAFTER RUNSTOP;SEQUENCE:MODE NUMACQS;NUMSEQUENCE 1;:ACQUIRE:NUMAVG 16;FASTACQ:STATE 0;PALETTE TEMPERATURE;:ACQUIRE:MODE SAMPLE;SAMPLINGMODE IT.

ACQUIRE:MAXSAMPLERATE? (Query Only)

This query returns the maximum real-time sample rate.

Group Acquisition

Syntax ACQUIRE:MAXSAMPLERATE?

Examples ACQUIRE:MAXSAMPLERATE? might return :ACQUIRE:MAXSAMPLERATE 2.5E9.

ACQUIRE:MODE

This command sets or queries the selected acquisition mode of the instrument.

Group Acquisition

Syntax ACQUIRE:MODE {SAMPLE|PEAKdetect|HIRES|AVERAGE|ENVELOPE}
ACQUIRE:MODE?

Related Commands [ACQUIRE:NUMAVg](#)
[CURVe?](#)

Arguments **SAMPle** specifies that the displayed data point value is the first sampled value that is taken during the acquisition interval. The [CURVe?](#) query, depending on sample rate, will result in either 8 bit or 16 bit data. In case of 8 bit data, the precision is also 8 bit. However, in case of 16 bit data, 12 bit precision data is zero padded in lower 4 bits. **SAMPle** is the default mode.

PEAKdetect specifies the display of high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. **PEAKdetect** mode can reveal the presence of aliasing or narrow spikes.

HIRes specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

AVERAGE specifies averaging mode, in which the resulting waveform shows an average of **SAMPle** data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the [ACQUIRE:NUMAVg](#) command.

ENVELOPE specifies envelope mode, where the resulting waveform displays the range of **PEAKdetect** from continued waveform acquisitions.

Examples **ACQUIRE:MODE ENVELOPE** sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

ACQUIRE:MODE? might return **:ACQUIRE:MODE AVERAGE**, indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

ACQUIRE:NUMACq? (Query Only)

This query-only command returns the number of waveform acquisitions that have occurred since the last time acquisitions were stopped.

Group Acquisition

Syntax **ACQUIRE:NUMACq?**

Related Commands [ACQUIRE:STATE](#)

Examples ACQUIRE:NUMACQ? might return :ACQUIRE:NUMACQ 350, indicating that 350 acquisitions have occurred since executing an ACQUIRE:STATE RUN command.

ACQUIRE:NUMAVG

This command sets or queries the number of waveform acquisitions that make up an averaged waveform. Ranges from 2 to 10240.

Group Acquisition

Syntax ACQUIRE:NUMAVG <NR1>
ACQUIRE:NUMAVG?

Related Commands [ACQUIRE:MODE](#)

Arguments <NR1> is the number of waveform acquisitions to average.

Examples ACQUIRE:NUMAVG 10 specifies that 10 waveform averages will be performed before exponential averaging starts.

ACQUIRE:NUMAVG? might return :ACQUIRE:NUMAVG 75, indicating that there are 75 acquisitions specified for averaging.

ACQUIRE:SEQUENCE:CURRENT? (Query Only)

In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.

Group Acquisition

Syntax ACQUIRE:SEQUENCE:CURRENT?

Examples ACQUIRE:SEQUENCE:CURRENT? might return :ACQUIRE:SEQUENCE:CURRENT 5.

ACQUIRE:SEQUENCE:MODE

In single sequence acquisition, the single sequence stop after count is based on the number of acquisitions.

Group	Acquisition
Syntax	ACQUIRE:SEQUENCE:MODE NUMACQS
Arguments	NUMACQS is the number of acquisitions.
Examples	ACQUIRE:SEQUENCE:MODE? might return :ACQUIRE:SEQUENCE:MODE NUMACQS indicating the acquisition sequence mode is set to NUMACQS.

ACQUIRE:SEQUENCE:NUMSEQUENCE

In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence. The default is 1.

Group	Acquisition
Syntax	ACQUIRE:SEQUENCE:NUMSEQUENCE <NR1> ACQUIRE:SEQUENCE:NUMSEQUENCE?
Arguments	<NR1> is the number of acquisitions or measurements that comprise the sequence.
Examples	ACQUIRE:SEQUENCE:NUMSEQUENCE 2 sets the number of acquisition in a sequence is set to 2. ACQUIRE:SEQUENCE:NUMSEQUENCE? might return ACQUIRE:SEQUENCE:NUMSEQUENCE 1, indicating that the number of acquisition in a sequence is set to 1.

ACQUIRE:STATE

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.

If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, a reset occurs and acquired data continues to acquire.

If acquire:stopafter is SEQUENCE, this command leaves the instrument in single sequence, unlike the run/stop button which takes the instrument out of single sequence.

Group Acquisition

Syntax ACQUIRE:STATE {<NR1>|OFF|ON|RUN|STOP}
ACQUIRE:STATE?

Related Commands [ACQUIRE:STOPAfter](#)

Arguments <NR1> = 0 stops acquisitions; any other value starts acquisitions.

OFF stops acquisitions.

ON starts acquisitions.

RUN starts acquisitions.

STOP stops acquisitions.

Examples ACQUIRE:STATE RUN starts the acquisition of waveform data and resets the count of the number of acquisitions.

ACQUIRE:STATE? might return :ACQUIRE:STATE 0, indicating that the acquisition is stopped.

ACQUIRE:STOPAfter

This command sets or queries whether the instrument continually acquires acquisitions or acquires a single sequence. Pressing SINGLE on the front panel button is equivalent to sending these commands: ACQUIRE:STOPAFTER SEQUENCE and ACQUIRE:STATE 1.

Group Acquisition

Syntax ACQUIRE:STOPAfter {RUNSTOP|SEQUENCE}
ACQUIRE:STOPAfter?

Related Commands [ACQUIRE:STATE](#)

Arguments RUNSTOP specifies that the instrument will continually acquire data, if ACQUIRE:STATE is turned on.
SEQUENCE specifies that the next acquisition will be a single-sequence acquisition.

Examples ACQUIRE:STOPAFTER RUNSTOP sets the instrument to continually acquire data.
ACQUIRE:STOPAFTER? might return :ACQUIRE:STOPAFTER SEQUENCE, indicating that the next acquisition the instrument makes will be of the single-sequence type.

ACTONEVENT:ENABLE

This command enables or disables actions on event (AOE). If AOE saves are limited and the limit has been reached

Group Act On Event

Syntax ACTONEVENT:ENABLE <NR1>
ACTONEVENT:ENABLE?

Arguments <NR1> is a number that enables or disables actions on event. The number zero disables the feature and the number one enables the feature.

Examples ACTONEVENT:ENABLE 1 enables actions on event.
ACTONEVENT:ENABLE? might return :ACTONEVENT:ENABLE 0, indicating the actions on event feature is disabled.

ACTONEVENT:LIMIT

This command sets whether the act on event should limit the number of saves. This prevents the saves from filling the hard drive.

Group Act On Event

Syntax ACTONEVENT:LIMIT <NR1>
ACTONEVENT:LIMIT?

Arguments <NR1> is a number that enables or disables whether the act on event should limit the number of saves. The number zero disables the feature and the number one enables the feature.

Examples ACTONEvent:LIMit 1 enables the limiting of act on event saves.
 ACTONEvent:LIMit? might return :ACTONEvent:LIMit 0, indicating the limiting of act on event saves feature is disabled.

ACTONEvent:LIMITCount

This command sets the limit of act on even saves. The ACTONEvent:LIMit command must be enabled.

Group Act On Event

Syntax ACTONEvent:LIMITCount <NR1>
 ACTONEvent:LIMITCount?

Arguments <NR1> sets the number of allowed saves for the act on event feature. The number must be a positive integer.

Examples ACTONEvent:LIMITCount 2 sets the number of saves to 2.
 ACTONEvent:LIMITCount? might return :ACTONEvent:LIMITCount 4, indicating the limit of act on event saves is 4.

ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE

This command saves a screen capture when a mask test fails.

Group Act On Event

Syntax ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE {ON|OFF|<NR1>}
 ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE?

Related Commands [SAVEONEvent:IMAGE:FILEFormat](#)
[SAVEONEvent:FILENAME](#)
[SAVEONEvent:WAVEform:FILEFormat](#)

[SAVEONEVent:WAVEform:SOUrce](#)

- Arguments** ON enables the save screen capture when a mask test fails feature.
OFF disables the save screen capture when a mask test fails feature.
<NR1> is a number that enables or disables the save screen capture when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.
- Examples** ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE ON sets the state of the save screen capture feature to on.
ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE? might return :ACTONEvent:MASKFail:ACTION:SAVEIMAGE:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test fails.

- Group** Act On Event
- Syntax** ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE?
- Related Commands** [SAVEONEVent:IMAGE:FILEFormat](#)
[SAVEONEVent:FILENAME](#)
[SAVEONEVent:WAVEform:FILEFormat](#)
[SAVEONEVent:WAVEform:SOUrce](#)
- Arguments** ON enables the save source waveform(s) when a mask test fails feature.
OFF disables the save source waveform(s) when a mask test fails feature.
<NR1> is a number that enables or disables the save source waveform(s) when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.
- Examples** ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE? might return :ACTONEvent:MASKFail:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEvent:MASKFail:ACTION:SRQ:STATE

This command generates an SRQ event when a mask fails.

Group Act On Event

Syntax ACTONEvent:MASKFail:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEvent:MASKFail:ACTION:SRQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments ON enables the generate an SRQ event when a mask fails feature.
OFF disables the generate an SRQ event when a mask fails feature.
<NR1> is a number that enables or disables the generate an SRQ event when a mask fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:MASKFail:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask fails feature to enabled.
ACTONEvent:MASKFail:ACTION:SRQ:STATE? might return :ACTONEvent:MASKFail:ACTION:SRQ:STATE 0, indicating the state of the generate an SRQ event when a mask fails feature is disabled.

ACTONEvent:MASKFail:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test fails.

Group Act On Event

Syntax ACTONEvent:MASKFail:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEvent:MASKFail:ACTION:STOPACQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments	<p>ON enables the stop acquisitions when a mask test fails feature.</p> <p>OFF disables the stop acquisitions when a mask test fails feature.</p> <p><NR1> is a number that enables or disables the stop acquisitions when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKFail:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.</p> <p>ACTONEvent:MASKFail:ACTION:STOPACQ:STATE? might return :ACTONEvent:MASKFail:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.</p>

ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE

This command saves a screen capture when a mask hit occurs.

Group	Act On Event
Syntax	ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE {ON OFF <NR1>} ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE?
Related Commands	SAVEONEvent:IMAGE:FILEFormat SAVEONEvent:FILENAME SAVEONEvent:WAVEform:FILEFormat SAVEONEvent:WAVEform:SOURce
Arguments	<p>ON enables the save screen capture when a mask hit occurs feature.</p> <p>OFF disables the save screen capture when a mask hit occurs feature.</p> <p><NR1> is a number that enables or disables the save screen capture when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE ON sets the state of the save screen capture feature to on.</p> <p>ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE? might return :ACTONEvent:MASKHit:ACTION:SAVEIMAGE:STATE OFF, indicating the state of the save screen capture feature is set to off.</p>

ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask hit occurs.

Group Act On Event

Syntax ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE?

Related Commands [SAVEONEvent:IMAGe:FILEFormat](#)
[SAVEONEvent:FILEName](#)
[SAVEONEvent:WAVEform:FILEFormat](#)
[SAVEONEvent:WAVEform:SOURce](#)

Arguments ON enables the save source waveform(s) when a mask hit occurs feature.
OFF disables the save source waveform(s) when a mask hit occurs feature.
<NR1> is a number that enables or disables the save source waveform(s) when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.
ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE? might return :ACTONEvent:MASKHit:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEvent:MASKHit:ACTION:SRQ:STATE

This command generates an SRQ event when a mask hit occurs.

Group Act On Event

Syntax ACTONEvent:MASKHit:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEvent:MASKHit:ACTION:SRQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments	<p>ON enables the generate an SRQ event when a mask hit occurs feature.</p> <p>OFF disables the generate an SRQ event when a mask hit occurs feature.</p> <p><NR1> is a number that enables or disables the generate an SRQ event when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKHit:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask hit occurs feature.</p> <p>ACTONEvent:MASKHit:ACTION:SRQ:STATE? might return :ACTONEvent:MASKHit:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a mask hit occurs feature is disabled.</p>

ACTONEvent:MASKHit:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask hit occurs.

Group	Act On Event
Syntax	ACTONEvent:MASKHit:ACTION:STOPACQ:STATE {ON OFF <NR1>} ACTONEvent:MASKHit:ACTION:STOPACQ:STATE?
Related Commands	ACTONEvent:ENable
Arguments	<p>ON enables the stop acquisitions when a mask hit occurs feature.</p> <p>OFF disables the stop acquisitions when a mask hit occurs feature.</p> <p><NR1> is a number that enables or disables the stop acquisitions when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKHit:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.</p> <p>ACTONEvent:MASKHit:ACTION:STOPACQ:STATE? might return :ACTONEvent:MASKHit:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.</p>

ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE

This command saves a screen capture when a mask test passes.

Group	Act On Event
Syntax	ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE {ON OFF <NR1>} ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE?
Related Commands	SAVEONEVent:IMAGE:FILEFormat SAVEONEVent:FILENAME SAVEONEVent:WAVEform:FILEFormat SAVEONEVent:WAVEform:SOURce
Arguments	ON enables the save screen capture when a mask test passes feature. OFF disables the save screen capture when a mask test passes feature. <NR1> is a number that enables or disables the save screen capture when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.
Examples	ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE ON sets the state of the save screen capture feature to on. ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE? might return :ACTONEvent:MASKPass:ACTION:SAVEIMAGE:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test passes.

Group	Act On Event
Syntax	ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE {ON OFF <NR1>} ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE?
Related Commands	SAVEONEVent:IMAGE:FILEFormat SAVEONEVent:FILENAME SAVEONEVent:WAVEform:FILEFormat SAVEONEVent:WAVEform:SOURce

Arguments	<p>ON enables the save source waveform(s) when a mask test passes feature.</p> <p>OFF disables the save source waveform(s) when a mask test passes feature.</p> <p><NR1> is a number that enables or disables the save source waveform(s) when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.</p> <p>ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE? might return :ACTONEvent:MASKPass:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.</p>

ACTONEvent:MASKPass:ACTION:SRQ:STATE

This command generates an SRQ event when a mask passes.

Group	Act On Event
Syntax	ACTONEvent:MASKPass:ACTION:SRQ:STATE {ON OFF <NR1>} ACTONEvent:MASKPass:ACTION:SRQ:STATE?
Related Commands	ACTONEvent:ENable
Arguments	<p>ON enables the generate an SRQ event when a mask passes feature.</p> <p>OFF disables the generate an SRQ event when a mask passes feature.</p> <p><NR1> is a number that enables or disables the generate an SRQ event when a mask passes feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKPass:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask passes feature.</p> <p>ACTONEvent:MASKPass:ACTION:SRQ:STATE? might return :ACTONEvent:MASKPass:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a mask passes feature is disabled.</p>

ACTONEvent:MASKPass:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test passes.

Group	Act On Event
Syntax	ACTONEvent:MASKPass:ACTION:STOPACQ:STATE {ON OFF <NR1>} ACTONEvent:MASKPass:ACTION:STOPACQ:STATE?
Related Commands	ACTONEvent:ENable
Arguments	<p>ON enables the stop acquisitions when a mask test passes feature.</p> <p>OFF disables the stop acquisitions when a mask test passes feature.</p> <p><NR1> is a number that enables or disables the stop acquisitions when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p>ACTONEvent:MASKPass:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.</p> <p>ACTONEvent:MASKPass:ACTION:STOPACQ:STATE? might return :ACTONEvent:MASKPass:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.</p>

ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE

This command saves a screen capture when the user-set measurement limit is exceeded.

Group	Act On Event
Syntax	ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE {ON OFF <NR1>} ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE?
Related Commands	SAVEONEvent:IMAGE:FILEFormat SAVEONEvent:FILENAME SAVEONEvent:WAVEform:FILEFormat SAVEONEvent:WAVEform:SOURce
Arguments	ON enables the save screen capture when the measurement limit is exceeded feature.

OFF disables the save screen capture when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save screen capture when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE ON` sets the state of the save screen capture feature to on.

`ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE?` might return `:ACTONEvent:MEASUREMENT:ACTION:SAVEIMAGE:STATE OFF`, indicating the state of the save screen capture feature is set to off.

ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when the user-set measurement limit is exceeded.

Group Act On Event

Syntax `ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE`
 `{ON|OFF|<NR1>}`
 `ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE?`

Related Commands [SAVEONEvent:IMAGE:FILEFormat](#)
 [SAVEONEvent:FILENAME](#)
 [SAVEONEvent:WAVEform:FILEFormat](#)
 [SAVEONEvent:WAVEform:SOURce](#)

Arguments ON enables the save source waveform(s) when the measurement limit is exceeded feature.

OFF disables the save source waveform(s) when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save source waveform(s) when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE? might return :ACTONEvent:MEASUREMENT:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE

This command generates an SRQ event when any measurement triggers the user-defined measurement limits.

Group Act On Event

Syntax ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments ON enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

OFF disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

<NR1> is a number that enables or disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE 1 enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE? might return :ACTONEvent:MEASUREMENT:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when any measurement triggers the user-defined measurement limits feature is disabled.

ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE

This command stops acquisitions when the user-set measurement limit is exceeded.

Group Act On Event

Syntax ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments ON enables the stop acquisitions when the measurement limit is exceeded feature.
OFF disables the stop acquisitions when the measurement limit is exceeded feature.
<NR1> is a number that enables or disables the stop acquisitions when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.
ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE? might return :ACTONEvent:MEASUREMENT:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE

This command saves a screen capture when a search event is found.

Group Act On Event

Syntax ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE {ON|OFF|<NR1>}
ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE?

Related Commands [SAVEONEvent:IMAGE:FILEFormat](#)
[SAVEONEvent:FILENAME](#)
[SAVEONEvent:WAVEform:FILEFormat](#)
[SAVEONEvent:WAVEform:SOURce](#)

Arguments ON enables the save screen capture when a search event is found feature.
OFF disables the save screen capture when a search event is found feature.
<NR1> is a number that enables or disables the save screen capture when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE ON` sets the state of the save screen capture feature to on.

`ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE?` might return
`:ACTONEvent:SEARCH:ACTION:SAVEIMAGE:STATE OFF`, indicating the state of the save screen capture feature is set to off.

ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a search event is found.

Group Act On Event

Syntax `ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}`
`ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE?`

Related Commands [SAVEONEvent:IMAGE:FILEFormat](#)
[SAVEONEvent:FILENAME](#)
[SAVEONEvent:WAVEform:FILEFormat](#)
[SAVEONEvent:WAVEform:SOURce](#)

Arguments `ON` enables the save source waveform(s) when a search event is found feature.
`OFF` disables the save source waveform(s) when a search event is found feature.
`<NR1>` is a number that enables or disables the save source waveform(s) when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE?` might return
`:ACTONEvent:SEARCH:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEvent:SEARCH:ACTION:SRQ:STATE

This command generates an SRQ event when any search event is found.

Group	Act On Event
Syntax	<code>ACTONEvent:SEARCH:ACTION:SRQ:STATE {ON OFF <NR1>}</code> <code>ACTONEvent:SEARCH:ACTION:SRQ:STATE?</code>
Related Commands	ACTONEvent:ENable
Arguments	<p>ON enables the generate an SRQ event when any search event is found feature.</p> <p>OFF disables the generate an SRQ event when any search event is found feature.</p> <p><NR1> is a number that enables or disables the generate an SRQ event when any search event is found feature. The number zero disables the feature, all other numbers enable the feature.</p>
Examples	<p><code>ACTONEvent:SEARCH:ACTION:SRQ:STATE 1</code> enables the generate an SRQ event when any search event is found feature.</p> <p><code>ACTONEvent:SEARCH:ACTION:SRQ:STATE?</code> might return <code>:ACTONEvent:SEARCH:ACTION:SRQ:STATE 0</code>, indicating the generate an SRQ event when any search event is found feature is disabled.</p>

ACTONEvent:SEARCH:ACTION:STOPACQ:STATE

This command stops acquisitions when a search event is found.

Group	Act On Event
Syntax	<code>ACTONEvent:SEARCH:ACTION:STOPACQ:STATE {ON OFF <NR1>}</code> <code>ACTONEvent:SEARCH:ACTION:STOPACQ:STATE?</code>
Related Commands	ACTONEvent:ENable
Arguments	<p>ON enables the stop acquisitions when a search event is found feature.</p> <p>OFF disables the stop acquisitions when a search event is found feature.</p> <p><NR1> is a number that enables or disables the stop acquisitions when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.</p>

Examples `ACTONEvent:SEARCH:ACTION:STOPACQ:STATE ON` sets the state of the stop acquisitions feature to on.

`ACTONEvent:SEARCH:ACTION:STOPACQ:STATE?` might return
`:ACTONEvent:SEARCH:ACTION:STOPACQ:STATE OFF`, indicating the state of the stop acquisitions feature is set to off.

ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE

This command saves a screen capture on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation. This command replaces [SAVe:IMAGe](#) (still valid command, but only an alias for this new command).

Group Act On Event

Syntax `ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}`
`ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE?`

Arguments `ON` enables the save screen capture on a trigger event feature.

`OFF` disables the save screen capture on a trigger event feature.

`<NR1>` is a number that enables or disables the save screen capture on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE ON` sets the state of the save screen capture feature to on.

`ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE?` might return
`:ACTONEvent:TRIGger:ACTION:SAVEIMAGe:STATE OFF`, indicating the state of the save screen capture feature is set to off.

ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation. This command replaces [SAVEON:WAVEform](#) (still valid command, but only an alias for this new command).

Group Act On Event

Syntax ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE?

Arguments ON enables the save source waveform(s) on a trigger event feature.
OFF disables the save source waveform(s) on a trigger event feature.
<NR1> is a number that enables or disables the save source waveform(s) on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.
ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE? might return :ACTONEvent:TRIGger:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEvent:TRIGger:ACTION:SRQ:STATE

This command generates an SRQ event when a trigger event occurs.

Group Act On Event

Syntax ACTONEvent:TRIGger:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEvent:TRIGger:ACTION:SRQ:STATE?

Related Commands [ACTONEvent:ENable](#)

Arguments ON enables the generate an SRQ event when a trigger event occurs feature.
OFF disables the generate an SRQ event when a trigger event occurs feature.
<NR1> is a number that enables or disables the generate an SRQ event when a trigger event occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEvent:TRIGger:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a trigger event occurs feature.
ACTONEvent:TRIGger:ACTION:SRQ:STATE? might return :ACTONEvent:TRIGger:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a trigger event occurs feature is disabled.

ACTONEvent:TRIGger:ACTION:STOPACQ:STATE

This command stops acquisitions on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation.

Group	Act On Event
Syntax	ACTONEvent:TRIGger:ACTION:STOPACQ:STATE {ON OFF <NR1>} ACTONEvent:TRIGger:ACTION:STOPACQ:STATE?
Arguments	ON enables the stop acquisitions on a trigger event feature. OFF disables the stop acquisitions on a trigger event feature. <NR1> is a number that enables or disables the stop acquisitions on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.
Examples	ACTONEvent:TRIGger:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on. ACTONEvent:TRIGger:ACTION:STOPACQ:STATE? might return :ACTONEvent:TRIGger:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

AFG:AMPLitude

Sets (or queries) the AFG amplitude in volts, peak to peak.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:AMPLitude <NR3> AFG:AMPLitude?
Arguments	<NR3> is a floating point number that represents the AFG amplitude, peak to peak, in volts.

Examples `AFG:AMPLITUDE 1.0` sets the AFG amplitude to 1.0 volts, peak to peak.
`AFG:AMPLITUDE?` might return `:AFG:AMPLITUDE 3.0000` indicating the amplitude is set to 3.0 Volts.

AFG:ARBitrary:SOUrce

This command sets or queries the source name for the Arbitrary Waveform. Currently supported sources are either waveform file (.wfm) or text file (.csv).

Conditions Requires option AFG.

Group AFG

Syntax `AFG:ARBitrary:SOUrce <QString>`
`AFG:ARBitrary:SOUrce?`

Arguments `<QString>` is the source name.

Examples `AFG:ARBITRARY:SOURCE "E:/Waveforms/Square.wfm"` sets the source waveform to `E:/Waveforms/Square.wfm`.
`AFG:ARBITRARY:SOURCE?` might return `"E:/Waveforms/Square.wfm"` indicating the source is set to `E:/Waveforms/Square.wfm`.

AFG:BURSt:CCOUnt

This command sets or returns the cycle count for AFG burst mode.

Conditions Requires option AFG.

Group AFG

Syntax `AFG:BURSt:CCOUnt <NR1>`
`AFG:BURSt:CCOUnt?`

Related Commands [AFG:BURSt:TRIGger](#)

Arguments `<NR1>` is the cycle count.

Examples `AFG:BURSt:CCOUnt 1000` sets the cycle count for AFG burst mode to 1000
`AFG:BURSt:CCOUnt?` might return `:AFG:BURSt:CCOUnt 50`, indicating that the cycle count for AFG burst mode is set to 50

AFG:BURSt:TRIGger (No Query Form)

This command triggers a burst on AFG output.

Conditions Requires option AFG.

Group AFG

Syntax `AFG:BURSt:TRIGger`

Related Commands [AFG:BURSt:CCOUnt](#)

Examples `AFG:BURSt:TRIGger` triggers a burst on AFG output.

AFG:FREQuency

Sets (or queries) the AFG frequency, in Hz.

Conditions Requires option AFG.

Group AFG

Syntax `AFG:FREQuency <NR3>`
`AFG:FREQuency?`

Arguments `<NR3>` is the floating point number that represents the AFG frequency, in Hz.

Examples `AFG:FREQUENCY 100.0E3` sets the AFG frequency to 100 kHz.
`AFG:FREQUENCY?` might return `:AFG:FREQUENCY 312.5000E+3` indicating the frequency is set to 312.5 kHz.

AFG:FUNCTION

Sets (or queries) which AFG function to execute.

Conditions Requires option AFG.

Group AFG

Syntax AFG:FUNCTION
 {SINE|SQUare|PULSe|RAMP|NOISE|DC|SINC|GAUSSian|LORENTz
 |ERISe|EDECAy|HAVERSINE|CARDIac|ARBitrary}
 AFG:FUNCTION?

Arguments

SINE
SQUare
PULSe
RAMP
NOISe
DC – The DC level is controlled by AFG:OFFSet .
SINC (Sin(x)/x)
GAUSSian
LORENTz
ERISe
EDECAy
HAVERSINE
CARDIac
ARBitrary

Examples AFG:FUNC LOREN specifies to generate the Lorentz function.

AFG:FUNCTION? might return :AFG:FUNCTION SINE indicating the AFG function is set to sine.

AFG:HIGHLLevel

This command sets (or queries) the high level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:HIGHLevel1 <NR3> AFG:HIGHLevel1?
Arguments	<NR3> is a floating point number that represents the AFG high level value, in volts.
Examples	AFG:HIGHLEVEL 1.0 sets the AFG high level value to 1.0 volts. AFG:HIGHLEVEL? might return :AFG:HIGHLEVEL 1.5000 indicating the high level is set to 1.5 Volts.

AFG:LOWLevel

This command sets (or queries) the low level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:LOWLevel1 <NR3> AFG:LOWLevel1?
Arguments	NR3 is the floating point number that represents the AFG low level value, in volts.
Examples	AFG:LOWLEVEL 1.0 sets the AFG low level value to 1.00 volts. AFG:LOWLEVEL? might return :AFG:LOWLEVEL -1.5000 indicating the low level is set to —1.5 Volts.

AFG:NOISEAdd:PERCent

Sets (or queries) the AFG additive noise level as a percentage. Minimum is 0.0%, maximum is 100.0% and increment is 1.0%.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:NOISEAdd:PERCent <NR3> AFG:NOISEAdd:PERCent?
Arguments	Floating point number that represents the AFG additive noise level, as a percentage.
Examples	AFG:NOISEADD:PERCENT 50 sets the AFG additive noise level to 50 percent. AFG:NOISEADD:PERCENT? might return :AFG:NOISEADD:PERCENT 0 indicating that no noise is added.

AFG:NOISEAdd:STATE

Sets (or queries) the AFG additive noise state.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:NOISEAdd:STATE {0 1 OFF ON} AFG:NOISEAdd:STATE?
Arguments	1 or ON turns on the AFG additive noise state. 0 or OFF turns it off.
Examples	AFG:NOISEADD:STATE ON turns on the additive noise state. AFG:NOISEADD:STATE? might return :AFG:NOISEADD:STATE 0 indicating the noise additive state is off.

AFG:OFFSet

Sets (or queries) the AFG offset value, in volts.

Conditions	Requires option AFG.
-------------------	----------------------

Group	AFG
Syntax	AFG:OFFSet <NR3> AFG:OFFSet?
Arguments	<NR3> is a floating point number that represents the AFG offset, in volts.
Examples	AFG:OFFSET 1.0 sets the AFG offset to 1.0 volts. AFG:OFFSET? might return AFG:OFFSET 0.0E+0 indicating there is no offset.

AFG:OUTPut:LOAd:IMPEDance

Sets (or queries) the AFG output load impedance.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:OUTPut:LOAd:IMPEDance {FIFty HIGHZ} AFG:OUTPut:LOAd:IMPEDance?
Arguments	FIFty sets the output load impedance to 50 Ohms. HIGHZ sets the output load impedance to the high-impedance state.
Examples	AFG:OUTP:LOA:IMPED FIF sets the AFG output load impedance to 50 Ohms. AFG:OUTPUT:LOAD:IMPEDANCE? might return AFG:OUTPUT:LOAD:IMPEDANCE HIGHZ indicating the load impedance is set to the high impedance state.

AFG:OUTPut:MODE

This command sets or returns the AFG output mode.

Conditions	Requires option AFG.
Group	AFG

Syntax AFG:OUTPut:MODE {OFF|CONTiNuOUS|BURSt}
AFG:OUTPut:MODE?

Related Commands [AFG:OUTPut:STATE](#)

Arguments OFF turns off the AFG output mode.
CONTiNuOUS turns the AFG output mode to continuous.
BURSt turns the AFG output mode to burst.

Examples AFG:OUTPut:MODE BURSt sets the AFG output mode to burst.
AFG:OUTPut:MODE? might return AFG:OUTPut:MODE CONTiNuOUS, indicating that the AFG output mode is set to continuous.

AFG:OUTPut:STATE

Sets (or queries) the AFG output state.

Conditions Requires option AFG.

Group AFG

Syntax AFG:OUTPut:STATE {0|1|OFF|ON}
AFG:OUTPut:STATE?

Arguments 1 or ON turns on the AFG output state.
0 or OFF turns it off.

Examples AFG:OUTPut:STATE ON turns on the AFG output state.
AFG:OUTPut:STATE? might return AFG:OUTPut:STATE 1 indicating the AFG output is on.

AFG:PERIod

Sets (or queries) the period of the AFG waveform, in seconds.

Conditions Requires option AFG.

Group	AFG
Syntax	AFG:PERIOD <NR3> AFG:PERIOD?
Arguments	NR3 is the floating point number that represents the AFG period value, in seconds.
Returns	The query response is returned in high precision NR3 format (up to 12 digits with more than 4 trailing 0 digits after the decimal point is omitted).
Examples	AFG:PERIOD 1 sets the AFG period value to 1 second. AFG:PERIOD? might return :AFG:PERIOD 3.2000E-6 indicating the AFG period is set to 3.2 μ s.

AFG:PULSE:WIDTH

Sets (or queries) the AFG pulse width, in seconds.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:PULSE:WIDTH <NR3> AFG:PULSE:WIDTH?
Arguments	NR3 is the floating point number that represents the pulse width, in seconds.
Examples	AFG:PULSE:WIDTH 100.0E-6 sets the AFG pulse width to 100 microseconds. AFG:PULSE:WIDTH? might return :AFG:PULSE:WIDTH 1.0000E-6 indicating the pulse width is set to 1 μ s.

AFG:RAMP:SYMMetry

Sets (or queries) the AFG ramp symmetry in percent. Minimum is 0.0%, maximum is 100.0% and increment is 0.10%.

Conditions	Requires option AFG.
-------------------	----------------------

Group	AFG
Syntax	AFG:RAMP:SYMMetry <NR3> AFG:RAMP:SYMMetry?
Arguments	Floating point number that represents the AFG ramp symmetry, as a percentage.
Examples	AFG:RAMP:SYMMETRY 50.0 sets the AFG ramp symmetry to 50 percent. AFG:RAMP:SYMMETRY? might return :AFG:RAMP:SYMMETRY 50.0000 indicating the symmetry is set to 50%.

AFG:SQUare:DUTy

Sets (or queries) the AFG duty cycle in percent. The minimum is 10.0%, maximum is 90.0% and increment is 0.10%.

Conditions	Requires option AFG.
Group	AFG
Syntax	AFG:SQUare:DUTy <NR3> AFG:SQUare:DUTy?
Arguments	Floating point number that represents the AFG duty cycle, as a percentage.
Examples	AFG:SQUARE:DUTY 50.0 sets the AFG duty cycle to 50 percent. AFG:SQUARE:DUTY? might return :AFG:SQUARE:DUTY 50.0000 indicating the duty cycle is set to 50%.

ALias

This command sets or queries the state of alias functionality, and it is identical to the ALIAS:STATE command.

Group	Alias
--------------	-------

Syntax ALIAS {OFF|ON|<NR1>}
ALIAS?

Related Commands [ALIAS:STATE](#)

Arguments OFF turns Alias expansion off.
ON turns Alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.
<NR1> = 0 disables Alias mode; any other value enables Alias mode.

Examples ALIAS ON turns the alias feature on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.
ALIAS? might return :ALIAS:STATE 1, indicating that the alias feature is on.

ALIAS:CATALOG? (Query Only)

This query-only command returns a list of the currently defined alias labels, separated by commas. If no aliases are defined, the query returns the string "".

Group Alias

Syntax ALIAS:CATALOG?

Examples ALIAS:CATALOG? might return the string :ALIAS:CATALOG "SETUP1", "TESTMENU1", "DEFAULT" showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

ALIAS:DEFINE

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALIAS:STATE is turned on. The query form of this command returns the definitions of a selected alias.

NOTE. Attempting to give two aliases the same name causes an error. To give a new alias the name of an existing alias, the existing alias must first be deleted.

Group Alias

Syntax	<pre>ALIAS:DEFine <QString><,>{<QString> <Block>} ALIAS:DEFine?</pre>
Related Commands	ALIAS:STATE
Arguments	<p>The first <QString> is the alias label.</p> <p>This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters.</p> <p>The second <QString> or <Block> is a complete sequence of program messages.</p> <p>The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.</p>
Examples	<pre>ALIAS:DEFINE "ST1",":RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE" defines an alias named "ST1" that recalls a previously saved setup and performs an autoset.</pre> <pre>ALIAS:DEFINE? "ST1" returns :ALIAS:DEFINE "ST1",#247 :RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE.</pre>

ALIAS:DELETE (No Query Form)

This command removes a specified alias and is identical to `ALIAS:DELETE:NAME`. An error message is generated if the named alias does not exist.

Group	Alias
Syntax	<code>ALIAS:DELETE <QString></code>
Related Commands	*ESR? ALIAS:DELETE:ALL
Arguments	<QString> is the name of the alias to be removed. Using <code>ALIAS:DELETE</code> without specifying an alias causes an execution error. <QString> must be a previously defined value.
Examples	<code>ALIAS:DELETE "SETUP1"</code> deletes the alias named SETUP1.

ALias:DELEte:ALL (No Query Form)

This command deletes all existing aliases.

Group Alias

Syntax ALIas:DELEte:ALL

Related Commands [ALias:DELEte](#)
[ALias:DELEte:NAME](#)

Examples ALIAS:DELETE:ALL deletes all existing aliases.

ALias:DELEte:NAME (No Query Form)

This command removes a specified alias. An error message is generated if the named alias does not exist. This command is identical to ALIas:DELEte.

Group Alias

Syntax ALIas:DELEte:NAME <QString>

Arguments <QString> is the name of the alias to remove. Using ALIas:DELEte:NAME without specifying an alias causes an exception error.<QString> must be a previously defined alias.

Examples ALIAS:DELETE:NAME "STARTUP" deletes the alias named STARTUP.

ALias:STATE

This command turns aliases on or off and is identical to the [ALias](#) command. The query form of this command returns the state of the aliases.

Group Alias

Syntax ALIas:STATE {<NR1>|OFF|ON}
ALIas:STATE?

Arguments <NR1> = 0 turns off aliases; any other value turns on aliases.
OFF turns alias expansion off.
ON turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples ALIAS:STATE OFF turns off the alias feature.
ALIAS:STATE? might return :ALIAS:STATE ON, indicating that alias expansion is currently turned on.

ALLEV? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the *ESR? query to enable the events to be returned. This command is similar to repeatedly sending *EVMsg? queries to the instrument.

Group Status and Error

Syntax ALLEV?

Related Commands [*ESR?](#)
[EVMsg?](#)

Examples ALLEV? might return :ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UNTERMINATED;".

AUTOSAVEPITIMEOUT

This command sets or queries the idle time from the programmable interface before auto-save occurs.

Group Miscellaneous

Syntax AUTOSAVEPITIMEOUT <NR1>
AUTOSAVEPITIMEOUT?

Arguments <NR1>

Examples `AUTOSAVEPITIMEOUT 100000` sets the timeout to 100 seconds.
`AUTOSAVEPITIMEOUT?` might return `:AUTOSAVEPITIMEOUT 180000`
 indicating the time out value is 180 seconds.

AUTOSAVEUITIMEOUT

This command sets or queries the idle time from the user interface before auto-save occurs.

Group Miscellaneous

Syntax `AUTOSAVEUITIMEOUT <NR1>`
`AUTOSAVEUITIMEOUT?`

Arguments `<NR1>`

Examples `AUTOSAVEUITIMEOUT 4000` sets the timeout to 4 seconds.
`AUTOSAVEUITIMEOUT?` might return `:AUTOSAVEUITIMEOUT 3000` indicating
 the time out value is 3 seconds.

AUTOSet (No Query Form)

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.

Group Miscellaneous

Syntax `AUTOSet EXECute`

Arguments `EXECute` autosets the displayed waveform; this is equivalent to pressing the front panel Autoset button.

Examples `AUTOSet EXECute` autosets the displayed waveform.

AUTOSet:ACQuisition:ENABle

This command sets or queries the Autoset acquisition setting adjustment. Settings affected may include, but not be limited to, acquisition mode, and FastAcq mode.

Group Miscellaneous

Syntax AUTOSet:ACQuisition:ENABle {ON|1|OFF|0}
AUTOSet:ACQuisition:ENABle?

Related Commands [AUTOSet:HORizontal:ENABle](#)
[AUTOSet:TRIGger:ENABle](#)
[AUTOSet:VERTical:ENABle](#)

Arguments ON or 1 enables Autoset to change acquisition settings.
OFF or 0 disables Autoset from changing acquisition settings.

Examples AUTOSet:ACQuisition:ENABle OFF turns off Autoset's adjustment of the acquisition settings.
AUTOSet:ACQuisition:ENABle? might return 0, indicating Autoset's adjustment of the acquisition settings is turned off.

AUTOSet:ENABle

This command sets or queries the Autoset enable/disable feature. This is useful for classroom purposes where the instructor wants the students to achieve the desired instrument settings without the benefit of the Autoset feature.

Group Miscellaneous

Syntax AUTOSet:ENABle {ON|1|OFF|0}
AUTOSet:ENABle?

Related Commands [AUTOSet:ACQuisition:ENABle](#)
[AUTOSet:HORizontal:ENABle](#)
[AUTOSet:TRIGger:ENABle](#)
[AUTOSet:VERTical:ENABle](#)

Arguments	ON or 1 enables Autoset. OFF or 0 disables Autoset.
Examples	AUTOSet:ENAbLe ON turns on the Autoset feature. AUTOSet:ENAbLe? might return 1, indicating that Autoset is enabled.

AUTOSet:HORizontal:ENAbLe

This command sets or queries Autoset's adjustment of horizontal settings. Settings affected may include, but not be limited to, horizontal scale, horizontal position, and horizontal delay mode.

Group Miscellaneous

Syntax AUTOSet:HORizontal:ENAbLe {ON|1|OFF|0}
AUTOSet:HORizontal:ENAbLe?

Related Commands [AUTOSet:ACQuisition:ENAbLe](#)
[AUTOSet:TRIGger:ENAbLe](#)
[AUTOSet:VERTical:ENAbLe](#)

Arguments ON or 1 enables Autoset to change horizontal settings.
OFF or 0 disables Autoset from changing horizontal settings.

Examples AUTOSet:HORizontal:ENAbLe OFF turns off Autoset's adjustment of the horizontal settings.
AUTOSet:HORizontal:ENAbLe? might return 0, indicating Autoset's adjustment of the horizontal settings is turned off.

AUTOSet:TRIGger:ENAbLe

This command sets or queries Autoset's adjustment of trigger settings. Settings affected may include, but not be limited to, trigger level, trigger source, and trigger coupling.

Group Miscellaneous

Syntax	<code>AUTOSet:TRIGger:ENAb1e {ON 1 OFF 0}</code> <code>AUTOSet:TRIGger:ENAb1e?</code>
Related Commands	AUTOSet:ACQuisition:ENAb1e AUTOSet:HORizontal:ENAb1e AUTOSet:VERTical:ENAb1e
Arguments	ON or 1 enables Autoset to change trigger settings. OFF or 0 disables Autoset from changing trigger settings.
Examples	<code>AUTOSet:TRIGger:ENAb1e OFF</code> turns off Autoset's adjustment of the trigger settings. <code>AUTOSet:TRIGger:ENAb1e?</code> might return 0, indicating Autoset's adjustment of the trigger settings is turned off.

AUTOSet:VERTical:ENAb1e

This command sets or queries Autoset's adjustment of vertical settings. Settings affected may include, but not be limited to, vertical scale, vertical position, and vertical offset.

Group	Miscellaneous
Syntax	<code>AUTOSet:VERTical:ENAb1e {ON 1 OFF 0}</code> <code>AUTOSet:VERTical:ENAb1e?</code>
Related Commands	AUTOSet:ACQuisition:ENAb1e AUTOSet:HORizontal:ENAb1e AUTOSet:TRIGger:ENAb1e
Arguments	ON or 1 enables Autoset to change vertical settings. OFF or 0 disables Autoset from changing vertical settings.
Examples	<code>AUTOSet:VERTical:ENAb1e OFF</code> turns off Autoset's adjustment of the vertical settings.

`AUTOSet:VERTical:ENABle?` might return 0, indicating Autoset's adjustment of the vertical settings is turned off.

AUTOSet:VERTical:OPTIMize

This command sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode (all waveforms are in one common graticule in the Waveform View).

Group Miscellaneous

Syntax `AUTOSet:VERTical:OPTIMize {RESOLution|VISIBility}`
`AUTOSet:VERTical:OPTIMize?`

Related Commands [DISplay:WAVEView<x>:VIEWStyle](#)

Arguments `RESOLution` uses as much of the ADC's (Analog to Digital Converter) range as possible to provide the best vertical resolution and measurement accuracy, but waveforms will overlap each other.

`VISIBility` vertically scales and positions waveforms so they are visually separated from each other at the expense of vertical resolution and measurement accuracy.

Examples `AUTOSet:VERTical:OPTIMize VISIBility` will vertically scale and position all active waveforms so that they are visually separated from each other when autoset is executed.

`AUTOSet:VERTical:OPTIMize?` might return `AUTOSet:VERTical:OPTIMize RESOLUTION`, which indicates that, when executed, autoset will vertically scale/position the active waveforms to maximize the ADC range while in Overlay display mode.

AUXout:EDGE

This command sets or queries the direction in which the Auxiliary Output signal will transition when a trigger occurs.

Group Miscellaneous

Syntax `AUXout:EDGE {RISing|FALling}`
`AUXout:EDGE?`

Arguments `RISing` sets the polarity to the rising edge.
`FALling` sets the polarity to the falling edge.

Examples `AUXOUT:EDGE RISING` sets the polarity to rising edge.
`AUXOUT:EDGE?` might return `:AUXOUT:EDGE FALLING`, indicating that the polarity is set to falling edge.

AUXout:SOURce

This command sets or queries the source at the AFG/AUX OUT, shared BNC connection.

Group Miscellaneous

Syntax `AUXout:SOURce {ATRIGger|OFF}`
`AUXout:SOURce?`

Arguments `ATRIGger` sets the source at the AFG/AUX OUT, shared BNC connector to the main trigger and turns off the AFG output (if on).
`OFF` turns off the output of the AFG/AUX OUT, shared BNC connector, this will turn off the AFG (if on).

Examples `AUXOUT:SOURCE?` might return `:AUXOUT:SOURCE ATRIGGER`, indicating that the source at the BNC connector is set to the A trigger.

BATTery:ACPOWER? (Query Only)

This command queries the state of AC power being plugged in.

Conditions Requires the battery pack accessory.

Group Battery

Syntax `BATTery:ACPOWER?`

Returns The command returns a 0 when no AC power is plugged in and a 1 when AC power is plugged in.

Examples BATTERY:ACPOWER? might return 0, indicating that no AC power is plugged into the instrument.

BATTERY:SLOT<1,2>:CHARGE? (Query Only)

This command queries the current charge of the battery.

Conditions Requires the battery pack accessory.

Group Battery

Syntax BATTERY:SLOT<1,2>:CHARGE?

Returns The command returns the charge percentage (0% to 100%). Returns NaN if there is no battery in the slot being queried.

Examples BATTERY:SLOT1:CHARGE? might return 75, indicating that the battery in slot 1 is 75% charged.

BATTERY:SLOT<1,2>:INSTALLED? (Query Only)

This command queries if a battery is installed.

Conditions Requires the battery pack accessory.

Group Battery

Syntax BATTERY:SLOT<1,2>:INSTALLED?

Returns The command returns 0 for battery not installed or 1 for battery installed.

Examples BATTERY:SLOT1:INSTALLED? might return 0, indicating that the battery is not installed.

BATTery:SLOT<1,2>:SERIALnumber? (Query Only)

This command queries the serial number of the battery.

Conditions Requires the battery pack accessory.

Group Battery

Syntax BATTery:SLOT<1,2>:SERIALnumber?

Returns The command returns the serial number of the battery (0 to 65534). Returns NaN if there is no battery in the slot being queried.

Examples BATTery:SLOT1:SERIALnumber? might return 10, indicating that the serial number of the battery in slot 1 is 10.

BATTery:SLOT<1,2>:TIMETOEMPTY? (Query Only)

This command queries the time to empty of the battery.

Conditions Requires the battery pack accessory.

Group Battery

Syntax BATTery:SLOT<1,2>:TIMETOEMPTY?

Returns The command returns the time, in minutes, until the battery in the slot being queried is depleted. Based on the current being pulled from the battery. NaN indicates that the battery being queried is not discharging or that there is no battery in the slot.

Examples BATTery:SLOT1:TIMETOEMPTY? might return 120, indicating that the battery in slot 1 has 2 hours until it is completely drained.

BATTery:SLOT<1,2>:TIMETOFULL? (Query Only)

This command queries the time to full of the battery.

Conditions	Requires the battery pack accessory.
Group	Battery
Syntax	BATTery:SLOT<1, 2>:TIMETOFULL?
Returns	The command returns the time, in minutes, until the battery in the slot being queried is charged. Based on the current being pushed into the battery. NaN can indicate that the battery being queried is not charging or that there is not battery in the slot.
Examples	BATTery:SLOT1:TIMETOFULL? might return 120, indicating that the battery in slot 1 has 2 hours until it is completely full.

BUS:ADDNew (No Query Form)

This command adds the specified bus. This command creates/adds the bus but does not display it (turn it on). In order to enable bus decoding and see the bus display on screen, send the [DISplay:WAVEView<x>:BUS:B<x>:STATE ON](#) command.

Group	Bus
Syntax	BUS:ADDNew <QString>
Related Commands	BUS:B<x>:TYPE DISplay:WAVEView<x>:BUS:B<x>:STATE
Arguments	<QString> is a quoted string of the form "B<NR1>" where NR1 is ≥1.
Examples	BUS:ADDNEW "B5"; :display:waveview1:bus:b5:state on creates bus 3 with the default type of Parallel, and then turns it on.

BUS:B<x>:CAN:BITRate

This command sets or queries the CAN bit rate. The bus number is specified by x. If you select Custom, use [BUS:B<x>:CAN:BITRate:VALue](#) to set the bit rate.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	BUS:B<x>:CAN:BITRate {RATE10K RATE100K RATE1M RATE125K RATE153K RATE20K RATE25K RATE250K RATE31K RATE33K RATE400K RATE50K RATE500K RATE62K RATE68K RATE800K RATE83K RATE92K CUSTom} BUS:B<x>:CAN:BITRate?
Related Commands	BUS:B<x>:CAN:BITRate:VALue
Arguments	B<x> is the number of the bus. Arguments specify the bit rate.
Examples	BUS:B1:CAN:BITRate RATE1M sets the CAN bit rate to 1 Mb. BUS:B1:CAN:BITRate? might return :BUS:B1:CAN:BITRATE RATE500K, indicating the CAN bit rate is set to 500 k.

BUS:B<x>:CAN:BITRate:VALue

This command sets or queries CAN bit rate when Custom is selected by [BUS:B<x>:CAN:BITRate](#). The bus number is specified by x.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	BUS:B<x>:CAN:BITRate:VALue <NR3> BUS:B<x>:CAN:BITRate:VALue?
Related Commands	BUS:B<x>:CAN:BITRate
Arguments	B<x> is the number of the bus. <NR3> specifies the CAN bit rate.

Returns <NR1> is the bit rate.

Examples `BUS:B1:CAN:BITRate:VALue 400.0E+3` sets the bit rate to 400 k.
`BUS:B1:CAN:BITRate:VALue?` might return `:BUS:B1:CAN:BITRATE:VALUE 500000`, indicating that the bit rate is set to 500,000.

BUS:B<x>:CAN:FD:BITRate

This command sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus. The bus is specified by x. If you select Custom, use `BUS:B<x>:CAN:FD:BITRate:CUSTom` to set the bit rate.

Conditions Requires option SR-AUTO.

Group Bus

Syntax `BUS:B<x>:CAN:FD:BITRate`
`{RATE1M|RATE2M|RATE3M|RATE4M|RATE5M|RATE6M|`
`RATE7M|RATE8M|RATE9M|RATE10M|RATE11M|RATE12M|RATE13M|`
`RATE14M|RATE15M|RATE16M|CUSTom}`
`BUS:B<x>:CAN:FD:BITRate?`

Related Commands [BUS:B<x>:CAN:FD:BITRate:CUSTom](#)

Arguments B<x> is the number of the bus.
Arguments specify the bit rate.

Examples `BUS:B1:CAN:BITRATE RATE1M` sets the CAN FD bit rate to 1 Mbps.
`BUS:B1:CAN:BITRATE?` might return `:BUS:B1:CAN:BITRATE RATE4M`, indicating that the CAN FD bit rate is 4 Mbps.

BUS:B<x>:CAN:FD:BITRate:CUSTom

This command sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group	Bus
Syntax	<code>BUS:B<x>:CAN:FD:BITRate:CUSTom <NR1></code> <code>BUS:B<x>:CAN:FD:BITRate:CUSTom?</code>
Related Commands	BUS:B<x>:CAN:BITRate
Arguments	<code>B<x></code> is the number of the bus. <code><NR1></code> is the custom FD bit rate for the specified bus.
Examples	<code>BUS:B1:CAN:FD:BITRATE:CUSTOM 1000000</code> sets the bit rate to 1,000,000. <code>BUS:B1:CAN:FD:BITRATE:CUSTOM?</code> might return <code>:BUS:B1:CAN:FD:BITRATE:CUSTOM 4000000</code> , indicating the bit rate is set to 4,000,000.

BUS:B<x>:CAN:SAMPLEpoint

This command sets or queries the sample point for the specified CAN bus. The bus is specified by x.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	<code>BUS:B<x>:CAN:SAMPLEpoint <NR1></code> <code>BUS:B<x>:CAN:SAMPLEpoint?</code>
Arguments	<code>B<x></code> is the number of the bus. <code><NR1></code> is the sample point, in percent, for the specified CAN bus.
Examples	<code>BUS:B1:CAN:SAMPLEpoint 40</code> sets the sample point to 40%. <code>BUS:B1:CAN:SAMPLEpoint?</code> might return <code>:BUS:B1:CAN:SAMPLEPOINT 50</code> indicating the sample point is at 50%.

BUS:B<x>:CAN:SIGNal

This command sets or queries the signal type for the specified CAN bus. The bus is specified by x.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	BUS:B<x>:CAN:SIGNal {DIFFerential CANH CANL RX TX} BUS:B<x>:CAN:SIGNal?
Arguments	B<x> is the number of the bus. Arguments are the CAN bus signal types.
Examples	BUS:B1:CAN:SIGNal CANL sets the signal type to CANL. BUS:B1:CAN:SIGNal? might return :BUS:B1:CAN:SIGNAL CANH indicating the signal type is CANH.

BUS:B<x>:CAN:SOUrce

This command sets or queries the CAN source channel for the specified CAN bus. The bus number is specified by x.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	BUS:B<x>:CAN:SOUrce {CH<x> DCH<x>_D<x> MATH<x> REF<x> REF<x>_D<x>} BUS:B<x>:CAN:SOUrce?
Arguments	B<x> is the number of the bus. CH<x> specifies an analog channel to use as the source. DCH<x>_D<x> specifies a digital channel to use as the source. The supported digital channel value is 1. The supported digital bits values are 0 to 15. MATH<x> specifies a math waveform to use as the source.

REF<x> specifies a reference waveform to use as the source.

REF<x>_D<x> specifies a digital reference waveform as the source.

Examples `BUS:B1:CAN:SOURce MATH6` sets the source channel to MATH6.
`BUS:B1:CAN:SOURce?` might return `:BUS:B1:CAN:SOURCE CH1`, indicating that the CAN source channel is CH1.

BUS:B<x>:CAN:STANDARD

This command sets or queries which CAN standard specification to analyze the specified CAN bus with. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax `BUS:B<x>:CAN:STANDARD {CAN2X|FDISO|FDNONISO}`
`BUS:B<x>:CAN:STANDARD?`

Arguments `B<x>` is the number of the bus.
`CAN2X` sets the CAN bus standard to CAN 2.0.
`FDISO` sets the CAN bus standard to ISO CAN FD (11898-1:2015).
`FDNONISO` sets the CAN bus standard to non-ISO CAN FD (Bosch:2012).

Examples `BUS:B1:CAN:STANDARD FDISO` sets the CAN standard to ISO CAN FD.
`BUS:B1:CAN:STANDARD?` might return `:BUS:B1:CAN:STANDARD CAN2X`, indicating that the CAN standard is CAN 2.0.

BUS:B<x>:CAN:THRESHOLD

This command sets or queries the source channel threshold for the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax	<code>BUS:B<x>:CAN:THRESHold <NR3></code> <code>BUS:B<x>:CAN:THRESHold?</code>
Arguments	<code>B<x></code> is the number of the bus. <code><NR3></code> is the source channel threshold for the specified CAN bus.
Examples	<code>BUS:B1:CAN:THRESHold 5</code> sets the threshold to 5 V. <code>BUS:B1:CAN:THRESHold?</code> might return <code>:BUS:B1:CAN:THRESHOLD 0.0E+0</code> indicating the threshold is set to 0.0 V.

BUS:B<x>:DISplay:FORMat

This command sets or queries how the data is represented in the bus form for the specified bus. The bus is specified by x.

Group	Bus
Syntax	<code>BUS:B<x>:DISplay:FORMat</code> { <code>HEX</code> <code>BINARY</code> <code>MIXEDASCII</code> <code>MIXEDHEX</code> <code>ASCII</code> <code>DECIMAL</code> <code>MIXED</code> } <code>BUS:B<x>:DISplay:FORMat?</code>
Arguments	<code>B<x></code> is the number of the bus.

NOTE. *Different bus types support only a subset of these arguments.*

HEX specifies hexadecimal format.

BINARY specifies binary format.

MIXEDASCII specifies mixed ASCII format.

MIXEDHEX specifies mixed hexadecimal format.

ASCII specifies ASCII format.

DECIMAL specifies decimal format.

MIXED specifies mixed format.

Examples	<code>BUS:B1:DISplay:FORMat BINARY</code> sets the format to binary. <code>BUS:B1:DISplay:FORMat?</code> might return <code>:BUS:B1:DISPLAY:FORMAT HEX</code> indicating the format is set to hexadecimal.
-----------------	---

BUS:B<x>:DISPlay:LAYout

This command sets or queries what to display for the specified bus. The bus is specified by x.

Group Bus

Syntax BUS:B<x>:DISPlay:LAYout {BUS|BUSANDWAVEFORM}
 BUS:B<x>:DISPlay:LAYout?

Arguments B<x> is the number of the bus.
 BUS specifies displaying the bus form only.
 BUSANDWAVEFORM specifies displaying the bus form and the constituent source waveform(s). This argument is not available for some bus types and some bus configurations

Examples BUS:B1:DISPlay:LAYout BUS specifies displaying the bus form only.
 BUS:B1:DISPlay:LAYout? might return :BUS:B1:DISPLAY:LAYOUT BUS indicating the bus will be displayed without displaying its constituent source waveform(s).

BUS:B<x>:I2C:CLOCK:SOURce

This command sets or queries the I2C clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax BUS:B<x>:I2C:CLOCK:SOURce {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}
 BUS:B<x>:I2C:CLOCK:SOURce?

Related Commands [BUS:B<x>:I2C:DATA:SOURce](#)
[BUS:B<x>:I2C:RWINADDR](#)

Arguments B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the I2C SCLK source.

DCH<x>_D<x> specifies a digital channel to use as the I2C SCLK source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> specifies a math waveform to use as the I2C SCLK source.

REF<x> specifies a reference waveform to use as the I2C SCLK source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified I2C bus.

Examples BUS:B1:I2C:CLOCK:SOURCE CH1 sets the I2C SCLK source to CH1.

BUS:B1:I2C:CLOCK:SOURCE? might return :BUS:B1:I2C:CLOCK:SOURCE CH4, indicating that the I2C SCLK source is set to channel 4.

BUS:B<x>:I2C:CLOCK:THReshold

This command sets or queries the I2C Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax BUS:B<x>:I2C:CLOCK:THReshold <NR3>
 BUS:B<x>:I2C:CLOCK:THReshold?

Arguments B<x> is the number of the bus.

<NR3> is the I2C Clock (SCLK) source threshold for the specified bus.

Examples BUS:B1:I2C:CLOCK:THReshold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:I2C:CLOCK:THReshold? might return :BUS:B1:I2C:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0 V.

BUS:B<x>:I2C:DATA:SOURce

This command sets or queries the I2C data (SDA) source for the specified I2C bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus
Syntax	BUS:B<x>:I2C:DATA:SOURce {CH<x> DCH<x>_D<x> MATH<x> REF<x> REF<x>_D<x>} BUS:B<x>:I2C:DATA:SOURce?
Related Commands	BUS:B<x>:I2C:CLOCK:SOURce BUS:B<x>:I2C:RWINADDR
Arguments	B<x> is the number of the bus. CH<x> specifies an analog channel to use as the I2C SDA source. DCH<x>_D<x> specifies a digital channel to use as the I2C SDA source. The supported digital channel value is 1. The supported digital bit values are 0 to 15. MATH<x> specifies a math waveform to use as the I2C SDA source. REF<x> specifies a reference waveform to use as the I2C SDA source. REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified I2C bus.
Examples	BUS:B1:I2C:DATA:SOURCE CH1 sets the I2C SDA source to CH1. BUS:B1:I2C:DATA:SOURCE? might return :BUS:B1:I2C:DATA:SOURCE MATH1, indicating that the I2C SDA source is set to MATH1.

BUS:B<x>:I2C:DATA:THReshold

This command sets or queries the I2C Data (SDA) source threshold for the specified bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus

Syntax	<code>BUS:B<x>:I2C:DATA:THResho1d <NR3></code> <code>BUS:B<x>:I2C:DATA:THResho1d?</code>
Arguments	<code>B<x></code> is the number of the bus. <code><NR3></code> is the I2C Data (SDA) source threshold for the specified bus.
Examples	<code>BUS:B1:I2C:DATA:THResho1d 50.0e-3</code> sets the threshold to 50 mV. <code>BUS:B1:I2C:DATA:THResho1d?</code> might return <code>:BUS:B1:I2C:DATA:THRESHOLD 0.0E+0</code> indicating the threshold is set to 0 V.

BUS:B<x>:I2C:RWINADDR

This command sets or queries the manner in which seven-bit I2C addresses are represented in the busform display of the specified bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus
Syntax	<code>BUS:B<x>:I2C:RWINADDR {0 1}</code> <code>BUS:B<x>:I2C:RWINADDR?</code>
Related Commands	BUS:B<x>:I2C:CLOCK:SOUrce BUS:B<x>:I2C:DATA:SOUrce
Arguments	<code>B<x></code> is the number of the bus. 0 displays seven-bit slave addresses as integers in the range of 0 to 127, with the state of the R/W* bit from the LSB of the slave address byte. For example, the slave address byte of 0b10100101 is displayed as the value 0x52 R. 1 displays the entire slave address byte as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1. For example, the slave address byte of 0b10100101 is displayed as the value 0xA5 R.
Examples	<code>BUS:B1:I2C:RWINADDR 0</code> displays seven-bit slave addresses as integers in the range of 0 to 127.

`BUS:B1:I2C:RWINADDR?` might return `:BUS:B1:I2C:RWINADDR 1`, indicating that the entire slave address byte is displayed as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1.

BUS:B<x>:LABel:COLor

This command sets or queries the color of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABel:COLor <QString>`
`BUS:B<x>:LABel:COLor?`

Arguments `B<x>` is the number of the bus.
`<QString>` is the bus label color. To return the color to the default color, send an empty string as in this example: `:BUS:B1:LABEL:COLOR ""`.

Examples `BUS:B1:LABel:COLor "#FF0000"` sets the label color to red.
`BUS:B1:LABel:COLor?` might return `:BUS:B1:LABEL:COLOR "#FF0000"` indicating the color is red.

BUS:B<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABel:FONT:BOLD {ON|OFF|1|0}`
`BUS:B<x>:LABel:FONT:BOLD?`

Arguments `B<x>` is the number of the bus.
`ON` displays the label in bold font.
`OFF` does not display the label in bold font.
`1` displays the label in bold font.

0 does not display the label in bold font.

Examples `BUS:B1:LABe1:FONT:BOLD OFF` turns off bold font.

`BUS:B1:LABe1:FONT:BOLD?` might return `:BUS:B1:LABEL:FONT:BOLD 1` indicating a bold font.

BUS:B<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABe1:FONT:ITALic {ON|OFF|1|0}`
`BUS:B<x>:LABe1:FONT:ITALic?`

Arguments B<x> is the number of the bus.
 ON displays the label in italic font.
 OFF does not display the label in italic font.
 1 displays the label in italic font.
 0 does not display the label in italic font.

Examples `BUS:B1LABe1:FONT:ITALic OFF` turns off italic font.

`BUS:B1LABe1:FONT:ITALic?` might return `:BUS:B1:LABEL:FONT:ITALIC 1` indicating the font is italic.

BUS:B<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABe1:FONT:SIZE <NR1>`
`BUS:B<x>:LABe1:FONT:SIZE?`

Arguments `B<x>` is the number of the bus.
 `<NR1>` is the font size.

Examples `BUS:B1:LABE1:FONT:SIZE 10` sets the font size to 10 points.
`BUS:B1:LABE1:FONT:SIZE?` might return `:BUS:B1:LABEL:FONT:SIZE 20`
 indicating the font size is 20 points.

BUS:B<x>:LABE1:FONT:TYPE

This command sets or queries the font type of the specified bus label, such as Arial or Times New Roman. The bus is specified by `x`.

Group Bus

Syntax `BUS:B<x>:LABE1:FONT:TYPE <QString>`
 `BUS:B<x>:LABE1:FONT:TYPE?`

Arguments `B<x>` is the number of the bus.

 `<QString>` is the specified font type. Available fonts include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples `BUS:B1:LABE1:FONT:TYPE Monospace` selects a monospace font.
`BUS:B1:LABE1:FONT:TYPE?` might return `:BUS:B1:LABEL:FONT:TYPE`
 `"Frutiger LT Std 55 Roman"` indicating the font type is Frutiger LT Std.

BUS:B<x>:LABE1:FONT:UNDERline

This command sets or queries the underline state of the specified bus label. The bus is specified by `x`.

Group Bus

Syntax `BUS:B<x>:LABE1:FONT:UNDERline {ON|OFF|1|0}`
 `BUS:B<x>:LABE1:FONT:UNDERline?`

Arguments B<x> is the number of the bus.
 ON displays the label in underlined font.
 OFF does not display the label in underlined font.
 1 displays the label in underlined font.
 0 does not display the label in underlined font.

Examples BUS:B:LABe1:FONT:UNDERline ON turns on underline font.
 BUS:B:LABe1:FONT:UNDERline? might return
 :BUS:B1:LABEL:FONT:UNDERLINE 0 indicating underline is off.

BUS:B<x>:LABel:name

This command sets or queries the label for the specified bus. The bus is specified by x.

Group Bus

Syntax BUS:B<x>:LABel:name <QString>
 BUS:B<x>:LABel:name?

Related Commands [BUS:B<x>:TYPe](#)

Arguments B<x> is the number of the bus.
 <QString> is an alphanumeric string of text enclosed in quotes. The text string is limited to 30 characters. It contains the text label information for the bus.

Examples BUS:B1:LABEL:NAME "TEST" sets the waveform label for the bus B1 to Test.
 BUS:B1:LABEL:NAME? might return :BUS:B1:LABEL:NAME "BUS 1",
 indicating that the waveform label for the bus B1 is set to "Bus 1".

BUS:B<x>:LABel:XPOS

This command sets or queries the x-position of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABe1:XPOS <NR3>`
 `BUS:B<x>:LABe1:XPOS?`

Arguments `B<x>` is the number of the bus.
 `<NR3>` is the x-position, in pixels relative to the left edge of the screen of the specified bus label.

Examples `BUS:B1:LABe1:XPOS 90` set the x position to 90.
 `BUS:B1:LABe1:XPOS?` might return `:BUS:B1:LABE1:XPOS 45.0000` indicating the x position is 45.0 pixels to the right of the left edge of the display.

BUS:B<x>:LABe1:YPOS

This command sets or queries the y-position of the specified bus label. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:LABe1:YPOS <NR3>`
 `BUS:B<x>:LABe1:YPOS?`

Arguments `B<x>` is the number of the bus.
 `<NR3>` is the y-position, in pixels relative to the baseline of the waveform, of the specified bus label.

Examples `BUS:B1:LABe1:YPOS 1.0e1` set the y position to 10.0.
 `BUS:B1:LABe1:YPOS?` might return `:BUS:B1:LABE1:YPOS 0.0E+0` indicating the y position is 0.0 pixels from the baseline of the waveform.

BUS:B<x>:LIN:BITRate

This command sets or queries the LIN bus bit rate. The bus number is specified by x. If you select Custom, use `BUS:B<x>:LIN:BITRate:CUSTom` to set the bit rate.

Conditions Requires option SR-AUTO.

Group Bus

Syntax `BUS:B<x>:LIN:BITRate`
`{RATE10K|RATE1K|RATE19K|RATE2K|RATE4K|RATE9K|CUSTom}`
`BUS:B<x>:LIN:BITRate?`

Related Commands [BUS:B<x>:LIN:BITRate:CUSTom](#)

Arguments `B<x>` is the number of the bus.
Arguments are the available bit rates.

Examples `BUS:B1:LIN:BITRate Rate4k` sets the bit rate to 4 kb.
`BUS:B1:LIN:BITRate?` might return `:BUS:B1:LIN:BITRATE RATE19K`,
indicating that the bit rate is set to 19 kb.

BUS:B<x>:LIN:BITRate:CUSTom

This command sets or queries LIN custom bit rate for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax `BUS:B<x>:LIN:BITRate:CUSTom <NR1>`
`BUS:B<x>:LIN:BITRate:CUSTom?`

Related Commands [BUS:B<x>:LIN:BITRate](#)

Arguments `B<x>` is the number of the bus.
`<NR1>` is the LIN custom bit rate for the specified bus.

Examples `BUS:B1:LIN:BITRate:CUSTom 9000` sets the bit rate to 9,000.
`BUS:B1:LIN:BITRate:CUSTom?` might return
`:BUS:B1:LIN:BITRate:CUSTom 10000` indicating the bit rate is set to 10,000.

BUS:B<x>:LIN:IDFOrmat

This command sets or queries LIN bus identifier format for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax BUS:B<x>:LIN:IDFOrmat {NOPARity|PARity}
BUS:B<x>:LIN:IDFOrmat?

Arguments B<x> is the number of the bus.
NOPARity specifies an id format that includes parity.
PARity specifies an id format that separates parity.

Examples BUS:B1:LIN:IDFOrmat PARITY set the id format to parity.
BUS:B1:LIN:IDFOrmat? might return :BUS:B1:LIN:IDFOrmat NOPARITY,
indicating the id format is set to no parity.

BUS:B<x>:LIN:POLarity

This command sets or queries the LIN source polarity for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax BUS:B<x>:LIN:POLarity {INVerted|NORmal}
BUS:B<x>:LIN:POLarity?

Arguments B<x> is the number of the bus.
INVerted specifies inverted polarity.
NORmal specifies normal polarity.

Examples `BUS:B1:LIN:POLarity Inverted` sets the polarity to inverted.
`BUS:B1:LIN:POLarity?` might return `:BUS:B1:LIN:POLARITY NORMAL`, indicating the bus polarity is set to normal.

BUS:B<x>:LIN:SAMPLEpoint

Specifies the LIN sample point, for the specified LIN bus. The bus is specified by *x*.

Conditions Requires option SR-AUTO.

Group Bus

Syntax `BUS:B<x>:LIN:SAMPLEpoint <NR1>`
`BUS:B<x>:LIN:SAMPLEpoint?`

Arguments `B<x>` is the number of the bus.
`<NR1>` is a percentage that represents the point at which to sample during each bit period.

Examples `BUS:B1:LIN:SAMPLEPOINT 10` sets the sample point to 10% of the bit period
`BUS:B1:LIN:SAMPLEPOINT?` might return `BUS:B1:LIN:SAMPLEPOINT 50` indicating that the sample point is set to 50% of the bit period

BUS:B<x>:LIN:SOURce

This command sets or queries the LIN data source for the specified bus. The bus number is specified by *x*.

Conditions Requires option SR-AUTO.

Group Bus

Syntax `BUS:B<x>:LIN:SOURce {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}`
`BUS:B<x>:LIN:SOURce?`

Arguments **B<x>** is the number of the bus.

CH<x> specifies an analog channel to use as the LIN data source.

DCH<x>_D<x> specifies a digital channel to use as the LIN data source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> specifies a math waveform to use as the LIN data source.

REF<x> specifies a reference waveform to use as the LIN data source.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform.

Examples **BUS:B1:LIN:SOURce MATH4** sets the source to MATH 4

BUS:B1:LIN:SOURce? might return **:BUS:B1:LIN:SOURCE CH1**, indicating the source is set to channel 1.

BUS:B<x>:LIN:SOURce:THReshold

This command sets or queries the LIN source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax **BUS:B<x>:LIN:SOURce:THReshoId <NR3>**
BUS:B<x>:LIN:SOURce:THReshoId?

Arguments **B<x>** is the number of the bus.

<NR3> the LIN source threshold for the specified bus.

Examples **BUS:B1:LIN:SOURce:THReshoId 50.0e-3** sets the threshold to 50 mV.

BUS:B1:LIN:SOURce:THReshoId? might return
:BUS:B1:LIN:SOURCE:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

BUS:B<x>:LIN:STANDard

This command sets or queries the LIN bus standard for the specified bus. The bus number is specified by x.

Conditions	Requires option SR-AUTO.
Group	Bus
Syntax	BUS:B<x>:LIN:STANDARD {MIXed V1X V2X} BUS:B<x>:LIN:STANDARD?
Arguments	B<x> is the number of the bus. MIXed specifies both versions 1.x and 2.x of the LIN standard. V1X specifies version 1.x of the LIN standard. V2X specifies version 2.x of the LIN standard.
Examples	BUS:B1:LIN:STANDARD V1X sets the standard to version 1.x. BUS:B1:LIN:STANDARD? might return :BUS:B1:LIN:STANDARD V2X, indicating the standard is set to version 2.x for bus B1.

BUS:B<x>:PARAllel:ALLTHResholds

This command sets or queries a threshold value for sources for the parallel bus. Use the [BUS:B<x>:PARAllel:ALLTHResholds:APPLY](#) command to set the thresholds to this value. The bus is specified by x.

Group	Bus
Syntax	BUS:B<x>:PARAllel:ALLTHResholds <NR3>
Related Commands	BUS:B<x>:PARAllel:ALLTHResholds:APPLY
Arguments	B<x> is the number of the bus. <NR3> is the source threshold.
Examples	BUS:B4:PARALLEL:ALLTHRESHOLDS 1.0 sets the threshold of all the sources in parallel Bus 4 to 1.0 V. BUS:B3:PARALLEL:ALLTHRESHOLDS? might return BUS:B3:PARALLEL:ALLTHRESHOLDS 500.00000E-3 indicates the threshold for all sources in parallel Bus 3 is currently set to 500.0 mV.

BUS:B<x>:PARAllel:ALLTHResholds:APPLY (No Query Form)

This command sets all of the data source thresholds to the value set by [BUS:B<x>:PARAllel:ALLTHResholds](#) for the parallel bus. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:PARAllel:ALLTHResholds:APPLY`

Related Commands [BUS:B<x>:PARAllel:ALLTHResholds](#)

Examples `BUS:B1:PARAllel:ALLTHResholds:APPLY` sets all data source thresholds to the value set by [BUS:B<x>:PARAllel:ALLTHResholds](#).

BUS:B<x>:PARAllel:BIT<x>SOURCE

This command sets or queries the specified bit source for specified parallel bus. The bus is specified by x. The bit is specified by n and is an integer in the range of 1 to 64.

Group Bus

Syntax `BUS:B<x>:PARAllel:BIT<x>SOURCE {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>| NONE}`

Related Commands [BUS:B<x>:PARAllel:BIT<n>SOURCE:THReshold](#)

Arguments

- B<x> is the number of the bus.
- CH<x> is the specified bit source.
- DCH<x>_D<x> specifies a digital channel to use as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
- MATH<x> is the specified bit source.
- REF<x> is the specified bit source.
- REF<x>_D<x> specifies a digital reference waveform as the bit<x> source waveform for the specified parallel bus.
- NONE disables the bit source.

Examples `BUS:B1:PARAllel:BIT1Source CH1` sets the bit 1 source to channel 1.
 `BUS:B1:PARAllel:BIT1Source?` might return
 `:BUS:B1:PARALLEL:BIT1SOURCE CH1` indicating the bit1 source is CH1.

BUS:B<x>:PARAllel:BIT<n>SOURce:THReshold

This command sets or queries the specified bit source threshold for the specified parallel bus. The bus is specified by x. The bit is specified by n and is an integer in the range of 1 to 64.

Group Bus

Syntax `BUS:B<x>:PARAllel:BIT<n>SOURce:THReshold <NR3>`

Related Commands [BUS:B<x>:PARAllel:BIT<x>SOURce](#)

Arguments `B<x>` is the number of the bus.
 `<NR3>` is the specified bit source threshold for the specified parallel bus.

Examples `BUS:B3:PARALLEL:BIT2SOURCE:THRESHOLD 1.0` sets the threshold of bit source 2 of parallel Bus 3 to 1.0 V.
 `BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD?` might return
 `:BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD 250.00000E-3` indicates the threshold of bit source 4 of parallel Bus 1 is currently set to 250.0 mV.

BUS:B<x>:PARAllel:CLOCK:EDGE

This command sets or queries the clock edge for the parallel bus. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:PARAllel:CLOCK:EDGE {FALLING|RISING|EITHER}`
 `BUS:B<x>:PARAllel:CLOCK:EDGE?`

Related Commands [BUS:B<x>:PARAllel:CLOCKSOURCE](#)

Arguments	<p>B<x> is the number of the bus.</p> <p>FALLING decodes on the falling edge of the clocked parallel bus signal.</p> <p>RISING decodes on the rising edge of the clocked parallel bus signal.</p> <p>EITHER decodes on the rising or falling edge of the clocked parallel bus signal.</p>
Examples	<p>BUS:B1:PARALLEL:CLOCK:EDGE FALLING sets the decoding to happen on the falling edge of its clocked parallel bus B1.</p> <p>BUS:B1:PARALLEL:CLOCK:EDGE? might return :BUS:B1:PARALLEL:CLOCK:EDGE RISING, indicating that when bus B1 operates in Clocked Parallel mode, it samples new data values on the rising edge of its clock source signal.</p>

BUS:B<x>:PARAllel:CLOCK:ISCLOCKED

This command determines whether the bus operates in a clocked or asynchronous fashion. The bus is specified by x.

Group	Bus
Syntax	<pre>BUS:B<x>:PARAllel:CLOCK:ISCLOCKED {OFF ON NR1}
BUS:B<x>:PARAllel:CLOCK:ISCLOCKED?</pre>
Arguments	<p>B<x> is the number of the bus.</p> <p>OFF argument specifies an asynchronous bus.</p> <p>ON argument specifies a clocked bus.</p> <p><NR1> = 0 specifies an asynchronous bus; any other value specifies a clocked bus.</p>
Examples	<p>BUS:B1:PARAllel:CLOCK:ISCLOCKED 0 sets the bus to operate asynchronously.</p> <p>BUS:B1:PARAllel:CLOCK:ISCLOCKED? might return :BUS:B1:PARALLEL:CLOCK:ISCLOCKED 1 indicating the bus is clocked.</p>

BUS:B<x>:PARAllel:CLOCKSOURCE

This command sets or queries the Parallel clock bit source for the specified bus. The bus is specified by x.

Group	Bus
Syntax	<code>BUS:B<x>:PARAllel:CLOCKSOURCE {CH<x> DCH<x>_D<x> MATH<x> REF<x> REF<x>_D<x> NONE}</code> <code>BUS:B<x>:PARAllel:CLOCKSOURCE?</code>
Related Commands	BUS:B<x>:PARAllel:CLOCK:ISCLOCKED
Arguments	<p><code>B<x></code> is the number of the bus.</p> <p><code>CH<x></code> specifies an analog FlexChannel to use as the bus clock source.</p> <p><code>DCH<x>_D<x></code> specifies a digital channel on a specified digital channel to use as the bus clock source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.</p> <p><code>MATH<x></code> specifies the math channel to use as the bus clock source.</p> <p><code>REF<x></code> specifies the reference channel to use as the bus clock source.</p> <p><code>REF<x>_D<x></code> specifies a digital reference waveform as the clock source waveform for the specified parallel bus.</p> <p><code>NONE</code> specifies the reference channel to use as the bus clock source.</p>
Examples	<p><code>BUS:B1:PARALLEL:CLOCK:SOURCE CH3</code> sets the Parallel clock source for the bus B1 to channel 3.</p> <p><code>BUS:B1:PARALLEL:CLOCK:SOURCE?</code> might return <code>:BUS:B1:PARALLEL:CLOCK:SOURCE CH8</code>, indicating that the Parallel clock source for the bus B1 is set to channel 8.</p>

BUS:B<x>:PARAllel:CLOCKSOURCE:THReshold

This command sets or queries the clock source threshold for the parallel bus. The bus is specified by x.

Group	Bus
Syntax	<code>BUS:B<x>:PARAllel:CLOCKSOURCE:THReshold <NR3></code> <code>BUS:B<x>:PARAllel:CLOCKSOURCE:THReshold?</code>
Related Commands	BUS:B<x>:PARAllel:CLOCK:ISCLOCKED

- Arguments** <NR3> is the clock bit source threshold for the parallel bus.
- Examples** `BUS:B4:PARALLEL:CLOCKSOURCE:THRESHOLD 1.5` sets the threshold of the clock source of parallel Bus 4 to 1.5 V.
- `BUS:B3:PARALLEL:CLOCKSOURCE:THRESHOLD?` might return `:BUS:B3:PARALLEL:CLOCKSOURCE:THRESHOLD 750.00000E-3` indicates the threshold of the clock source of parallel Bus 3 is currently set to 750.0 mV.

BUS:B<x>:RS232C:BITRate

This command sets or queries the RS-232C bit rate for bus<x>, where the bus number is specified by x. If you select Custom, use [BUS:B<x>:RS232C:BITRate:CUSTom](#) to set the bit rate.

- Conditions** Requires option SR-COMP.
- Group** Bus
- Syntax** `BUS:B<x>:RS232C:BITRate`
{`CUSTOM`|`RATE300`|`RATE1K`|`RATE2K`|`RATE9K`|`RATE19K`|`RATE38K`|`RATE115K`|`RATE921K`}
`BUS:B<x>:RS232C:BITRate?`
- Arguments** B<x> is the number of the bus.
Arguments are the available bit rates.
- Examples** `BUS:B1:RS232C:BITRATE RATE9K` sets the bit rate for the RS-232C bus B1 to 9000 bits-per-second.
- `BUS:B1:RS232C:BITRATE?` might return `:BUS:B1:RS232C:BITRATE 9000`, indicating that the bit rate for the RS-232C bus B1 is set to 9000 bits-per-second.

BUS:B<x>:RS232C:BITRate:CUSTom

This command sets or queries the RS-232C custom bit rate for the specified bus. The bus is specified by x.

- Conditions** Requires option SR-COMP.

Group	Bus
Syntax	<code>BUS:B<x>:RS232C:BITRate:CUSTOM <NR1></code> <code>BUS:B<x>:RS232C:BITRate:CUSTOM?</code>
Related Commands	BUS:B<x>:RS232C:BITRate
Arguments	<code>B<x></code> is the number of the bus. <code><NR1></code> is the custom bit rate.
Examples	<code>BUS:B1:RS232C:BITRATE:CUSTOM 9500</code> sets the bit rate for the RS-232C bus B1 to 9500 bits-per-second. <code>BUS:B1:RS232C:BITRATE:CUSTOM?</code> might return <code>:BUS:B1:RS232C:BITRATE:CUSTOM 9500</code> , indicating that the bit rate for the RS-232C bus B1 is set to 9500 bits-per-second.

BUS:B<x>:RS232C:DATABits

This command sets or queries the RS-232C data width for bus<x>, where the bus number is specified by x.

Conditions	Requires option SR-COMP.
Group	Bus
Syntax	<code>BUS:B<x>:RS232C:DATABits {7 8 9}</code> <code>BUS:B<x>:RS232C:DATABits?</code>
Arguments	<code>B<x></code> is the number of the bus. <code><NR1></code> specifies the number of bits in the RS-232C data frame.
Examples	<code>BUS:B1:RS232C:DATABITS 8</code> sets the data bits to 8 for the RS-232C bus B1. <code>BUS:B5:RS232C:DATABITS?</code> might return <code>:BUS:B1:RS232C:DATABITS 7</code> , indicating that the data bits is set to 7 for the RS-232C bus B5.

BUS:B<x>:RS232C:DELIMiter

This command sets or queries the RS-232C string delimiter on bus <x>, where the bus number is specified by x. This command only applies when Packet view is turned On.

Conditions Requires option SR-COMP.

Group Bus

Syntax BUS:B<x>:RS232C:DELIMiter {NULL|CR|LF|SPace|XFF}
BUS:B<x>:RS232C:DELIMiter?

Related Commands [BUS:B<x>:RS232C:DISplaymode](#)

Arguments B<x> is the number of the bus.
NULL specifies NULL (0x00) delimiting value for a packet.
CR specifies CR (0x0D) delimiting value for a packet.
LF specifies LF (0x0A) delimiting value for a packet.
XFF specifies XFF (0xFF) delimiting value for a packet.
SPace specifies SPace delimiting value for a packet.

Examples BUS:B1:RS232C:DELIMITER LF sets the delimiter to 0x0A for the RS-232C bus B1.
BUS:B1:RS232C:DELIMITER? might return :BUS:B1:RS232C:DELIMITER NULL, indicating that the delimiter to 0x00 for the RS-232C bus B1.

BUS:B<x>:RS232C:DISplaymode

This command sets or queries the RS-232C display mode for the specified bus. The bus is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax `BUS:B<x>:RS232C:DISplaymode {FRame|PACKET}`
`BUS:B<x>:RS232C:DISplaymode?`

Related Commands [BUS:B<x>:RS232C:DELIMiter](#)

Arguments `B<x>` is the number of the bus.

Frame displays each frame as a single entity.

PACKET displays a group of frames terminated with a single frame defined by the `BUS:B<x>:RS232C:DELIMiter` command.

Examples `BUS:B1:RS232C:DISPLAYMODE FRAME` sets the display mode for the RS-232C bus B1 to Frame.

`BUS:B1:RS232C:DISPLAYMODE?` might return `:BUS:B1:RS232C:DISPLAYMODE PACKET`, indicating that the display mode for the RS-232C bus B1 is set to Packet.

BUS:B<x>:RS232C:PARity

This command sets or queries the RS-232C parity for bus <x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax `BUS:B<x>:RS232C:PARity {NONE|EVEN|ODD}`
`BUS:B<x>:RS232C:PARity?`

Arguments `B<x>` is the number of the bus.

NONE specifies no parity.

EVEN specifies even parity.

ODD specifies odd parity.

Examples `BUS:B1:RS232C:PARITY ODD` sets the parity for the RS-232C bus B1 to odd.

`BUS:B1:RS232C:PARITY?` might return `:BUS:B1:RS232C:PARITY NONE`, indicating that the display mode for the RS-232C bus B1 is set to none.

BUS:B<x>:RS232C:POLarity

This command sets or queries the RS-232C source polarity for bus <x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax BUS:B<x>:RS232C:POLarity {NORma1|INVERTed}
BUS:B<x>:RS232C:POLarity?

Arguments B<x> is the number of the bus.
NORma1 sets the RS-232C bus polarity to positive.
INVERTed sets the RS-232C bus polarity to negative.

Examples BUS:B1:RS232C:POLARITY INVERTed sets the polarity for the RS-232C bus B1 to Inverted.

BUS:B1:RS232C:POLARITY? might return :BUS:B1:RS232C:POLARITY NORMAL, indicating that the polarity for the RS-232C bus B1 is set to Normal.

BUS:B<x>:RS232C:SOUrce

This command sets or queries the RS-232C source for the bus, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax BUS:B<x>:RS232C:SOUrce {CH<x>| DCH<x>_D<x>| REF<x>| MATH<x>|
REF<x>_D<x>}
BUS:B<x>:RS232C:SOUrce?

Arguments B<x> is the number of the bus.
CH<x> specifies an analog channel to use as the RS-232C source.

DCH<x>_D<x> specifies a digital channel of a specified digital channel to use for the RS-232C source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> specifies a math channel to use for the RS-232C source.

REF<x> specifies a reference channel to use for the RS-232C source.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified RS-232C bus.

Examples `BUS:B1:RS232C:SOURCE CH1` sets the source for the RS-232C bus B1 to channel 1.

`BUS:B1:RS232C:SOURCE?` might return `:BUS:B1:RS232C:SOURCE CH1`, indicating that the source for the RS-232C bus B1 is CH1.

BUS:B<x>:RS232C:SOUrcE:THReshold

This command sets or queries the RS-232C source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax `BUS:B<x>:RS232C:SOUrcE:THResho1d <NR3>`
`BUS:B<x>:RS232C:SOUrcE:THResho1d?`

Arguments B<x> is the number of the bus.
 <NR3> is the RS-232C source threshold for the specified bus.

Examples `BUS:B1:RS232C:SOUrcE:THRESHOLD 50.0e-3` sets the threshold to 50 mV.
`BUS:B1:RS232C:SOUrcE:THRESHOLD?` might return
`:BUS:B1:RS232C:SOUrcE:THRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:SENT:CHANWidth

This command sets or queries SENT fast channel bit widths for the specified bus.

Conditions	Requires option SRAUTOSEN.
Group	Bus
Syntax	BUS:B<x>:SENT:CHANwidth {TWELVETwelve FOURTEENTen SIXTEENEight} BUS:B<x>:SENT:CHANwidth?
Related Commands	BUS:B<x>:SENT:NUMCHANnel
Arguments	B<x> is the number of the bus waveform. TWELVETwelve sets both Fast Channel 1 and Fast Channel 2 to 12 bits wide. FOURTEENTen sets Fast Channel 1 to 14 bits and Fast Channel 2 to 10 bits. SIXTEENEight sets Fast Channel 1 to 16 bits and Fast Channel 2 to eight bits.
Examples	BUS:B1:SENT:CHANWIDTH TWELVETWELVE sets the channel width of both Fast Channel 1 and Fast Channel 2 to twelve bits. BUS:B1:SENT:CHANWIDTH? might return :BUS:B1:SENT:CHANWIDTH SIXTEENEIGHT, which indicates that the Fast Channel 1 width is 16 and the Fast Channel 2 width is eight.

BUS:B<x>:SENT:NIBBLECount

This command sets or queries SENT data nibbles for the specified bus.

Conditions	Requires option SRAUTOSEN.
Group	Bus
Syntax	BUS:B<x>:SENT:NIBBLECount {THREE FOUR SIX} BUS:B<x>:SENT:NIBBLECount?
Arguments	B<x> is the number of the bus waveform. THREE specifies regular data with 3 nibbles FOUR specifies regular data with 4 nibbles SIX specifies regular data with 6 nibbles

Examples `BUS:B1:SENT:NIBBLECOUNT THREE` sets the data nibble count to three.
`BUS:B1:SENT:NIBBLECOUNT?` might return `:BUS:B1:SENT:NIBBLECOUNT SIX`, which indicates the nibble count is set to six.

BUS:B<x>:SENT:NUMCHANnel

This command sets or queries SENT fast data channels for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax `BUS:B<x>:SENT:NUMCHANnel {ONE|TWO}`
`BUS:B<x>:SENT:NUMCHANnel?`

Arguments `B<x>` is the number of the bus waveform.
`ONE` specifies a SENT bus with one fast channel.
`TWO` specifies a SENT bus with two fast channels.

Examples `BUS:B1:SENT:NUMCHANnel TWO` sets the number of SENT channels to two.
`BUS:B1:SENT:NUMCHANNEL?` might return `:BUS:B1:SENT:NUMCHANNEL ONE` to indicate that the number of channels is set to one.

BUS:B<x>:SENT:PAUSEPULSe

This command sets or queries SENT pause pulse for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax `BUS:B<x>:SENT:PAUSEPULSe {NO|YES}`
`BUS:B<x>:SENT:PAUSEPULSe?`

Arguments `B<x>` is the number of the bus waveform.
`NO` specifies no pause pulse.

YES specifies a pause pulse is used.

Examples `BUS:B1:SENT:PAUSEPULSE NO` sets the pause pulse value to indicate no pause pulse.
`BUS:B1:SENT:PAUSEPULSE?` might return `:BUS:B1:SENT:PAUSEPULSE YES`, which indicates the SENT bus is set to use pause pulses.

BUS:B<x>:SENT:POLARITY

This command sets or queries SENT Idle State signal polarity for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax `BUS:B<x>:SENT:POLARITY {INverted|NORma1}`
`BUS:B<x>:SENT:POLARITY?`

Arguments `B<x>` is the number of the bus waveform.
`INverted` specifies inverted polarity.
`NORma1` specifies normal polarity.

Examples `BUS:B1:SENT:POLARITY INVERTED` sets the idle state of the bus to inverted polarity.
`BUS:B1:SENT:POLARITY?` might return `:BUS:B1:SENT:POLARITY NORMAL` to indicate the bus idle state polarity is set to normal

BUS:B<x>:SENT:SLOW

This command sets or queries the SENT slow channel configuration for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax	BUS:B<x>:SENT:SLOW {None ENHANCED4 ENHANCED8 SHOrt} BUS:B<x>:SENT:SLOW?
Arguments	<p>B<x> is the number of the bus waveform.</p> <p>None specifies no slow channel configured.</p> <p>ENHANCED4 specifies Enhanced 4 slow channel configuration.</p> <p>ENHANCED8 specifies Enhanced 8 slow channel configuration.</p> <p>SHOrt specifies short slow channel configuration.</p>
Examples	<p>BUS:B1:SENT:SLOW SHOrt sets the SENT bus to use the short slow channel configuration.</p> <p>BUS:B1:SENT:SLOW? might return :BUS:B1:SENT:SLOW NONE to indicate that the SENT bus is set to have no slow channel.</p>

BUS:B<x>:SENT:SOUrce

This command sets or queries the SENT DATA source for the specified bus.

Conditions	Requires option SRAUTOSEN.
Group	Bus
Syntax	BUS:B<x>:SENT:SOUrce {CH<x> DCH<x>_D<x> Math<x> REF<x> REF<x>_D<x>} BUS:B<x>:SENT:SOUrce?
Arguments	<p>B<x> is the number of the bus waveform.</p> <p>CH<x> specifies an analog channel as the clock source waveform for the audio bus.</p> <p>DCH<x>_D<x> specifies the channel bit of a separate digital channel as the clock source waveform for the audio bus. The supported digital channel value is 1. The supported digital bit values are 0 to 15.</p> <p>Math<x> specifies a math waveform as the clock source waveform for the audio bus.</p> <p>REF<x> specifies a reference waveform as the clock source waveform for the audio bus</p>

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified audio bus.

Examples `BUS:B1:SENT:SOURCE CH1` sets channel 1 as the source for the SENT bus.
`BUS:B1:SENT:SOURCE?` might return `:BUS:B1:SENT:SOURCE CH1` indicating that the source is set to channel 1

BUS:B<x>:SENT:THRESHold

This command sets or queries the SENT DATA source threshold for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax `BUS:B<x>:SENT:THRESHold <NR3>`
`BUS:B<x>:SENT:THRESHold?`

Arguments `B<x>` is the number of the bus waveform.
`<NR3>` sets the data source threshold value in volts.

Examples `BUS:B1:SENT:THRESHOLD 500.0E-3` sets the Bus 1 source threshold to 500.0 mV.
`BUS:B1:SENT:THRESHOLD?` might return `:BUS:B1:SENT:THRESHOLD 2.5000` to indicate the Bus 1 source threshold is 2.5 V.

BUS:B<x>:SENT:TICKTIME

This command sets or queries the SENT bus Clock Tick parameter for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax	<code>BUS:B<x>:SENT:TICKTIME <NR3></code> <code>BUS:B<x>:SENT:TICKTIME?</code>
Arguments	<code>B<x></code> is the number of the bus waveform. <code><NR3></code> is the SENT clock tick time, in seconds.
Examples	<code>BUS:B1:SENT:TICKTIME 3.0E-6</code> sets the SENT bus to clock tick to 3 microseconds. <code>BUS:B1:SENT:TICKTIME?</code> might return <code>:BUS:B1:SENT:TICKTIME 10.0000E-6</code> to indicate the clock tick is set to 10 microseconds.

BUS:B<x>:SENT:TICKTOLerance

This command sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.

Conditions	Requires option SRAUTOSEN.
Group	Bus
Syntax	<code>BUS:B<x>:SENT:TICKTOLerance <NR3></code> <code>BUS:B<x>:SENT:TICKTOLerance?</code>
Arguments	<code>B<x></code> is the number of the bus waveform. <code><NR3></code> is the tick tolerance percentage.
Examples	<code>BUS:B1:SENT:TICKTOLerance 20.0</code> sets the tick tolerance to 20%. <code>US:B1:SENT:TICKTOLerance?</code> might return <code>:BUS:B1:SENT:TICKTOLerance 22.5</code> to indicate the tolerance is set to 22.5%.

BUS:B<x>:SPI:BITOrder

This command sets or queries the SPI bit order for the specified bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
-------------------	---

Group	Bus
Syntax	BUS:B<x>:SPI:BITOrder {LSB MSB} BUS:B<x>:SPI:BITOrder?
Related Commands	BUS:B<x>:SPI:DATA:SIZE
Arguments	<p>B<x> is the number of the bus waveform.</p> <p>LSB specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.</p> <p>MSB specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.</p>
Examples	<p>BUS:B1:SPI:BITORDER LSB sets each bit order for the SPI Bus 1 to LSB.</p> <p>BUS:B1:SPI:BITORDER? might return BUS:B1:SPI:BITORDER MSB, indicating that the bit order for the SPI Bus 1 is set to MSB.</p>

BUS:B<x>:SPI:CLOCK:POLarity

This command sets or queries the SPI clock (SCLK) source polarity for the specified bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus
Syntax	BUS:B<x>:SPI:CLOCK:POLarity {FALL RISE} BUS:B<x>:SPI:CLOCK:POLarity?
Related Commands	BUS:B<x>:SPI:CLOCK:SOUrce
Arguments	<p>B<x> is the number of the bus waveform.</p> <p>FALL sets the clock to the falling edge of the signal.</p> <p>RISE sets the clock to the rising edge of the signal.</p>

Examples `BUS:B1:SPI:CLOCK:POLARITY FALL` sets the SPI clock polarity to the falling edge for the Bus 1.

`BUS:B1:SPI:CLOCK:POLARITY?` might return `BUS:B1:SPI:CLOCK:POLARITY RISE`, indicating that the SPI clock polarity for the Bus 1 is set to the rising edge.

BUS:B<x>:SPI:CLOCK:SOURce

This command sets or queries the SPI clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax `BUS:B<x>:SPI:CLOCK:SOURce {CH<x> | DCH<x>_D<x> | MATH<x> | REF<x> | REF<x>_D<x>}`
`BUS:B<x>:SPI:CLOCK:SOURce?`

Related Commands [BUS:B<x>:SPI:CLOCK:POLarity](#)

Arguments B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the bus SPI clock source.

DCH<x>_D<x> specifies the digital channel as the bus SPI clock source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> designates a math waveform as the clock source.

REF<x> designates a reference waveform as the clock source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified SPI bus.

Examples `BUS:B1:SPI:CLOCK:SOURCE CH5` sets the SPI clock source for the Bus 1 to CH5.

`BUS:B1:SPI:CLOCK:SOURCE?` might return `BUS:B1:SPI:CLOCK:SOURCE MATH1`, indicating that the SPI clock source for the Bus 1 is set to MATH1.

BUS:B<x>:SPI:CLOCK:THReshold

This command sets or queries the SPI Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax BUS:B<x>:SPI:CLOCK:THReshold <NR3>
BUS:B<x>:SPI:CLOCK:THReshold?

Arguments B<x> is the number of the bus waveform.
<NR3> is the SPI Clock (SCLK) source threshold for the specified bus.

Examples BUS:B1:SPI:CLOCK:THReshold 50.0e-3 sets the threshold to 50.0 mV.
BUS:B1:SPI:CLOCK:THReshold? might return
BUS:B1:SPI:CLOCK:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:DATA:POLarity

This command sets or queries the SPI Data (Data) source polarity for the bus number specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax BUS:B<x>:SPI:DATA:POLarity {HIGH|LOW}
BUS:B<x>:SPI:DATA:POLarity?

Arguments B<x> is the number of the bus waveform.
HIGH sets the SPI data polarity to active high.
LOW sets the SPI data polarity to active low.

- Examples** `BUS:B4:SPI:DATA:POLARITY HIGH` sets the SPI data polarity for the Bus 4 to active high.
- `BUS:B1:SPI:DATA:POLARITY?` might return `BUS:B1:SPI:DATA:POLARITY LOW`, indicating that the SPI SDA polarity for the Bus 1 is set to active low.

BUS:B<x>:SPI:DATA:SIZE

This command sets or queries the number of bits per word for the specified SPI bus. The bus is specified by *x*.

- Conditions** Requires option 2-SERIAL or 2-ULTIMATE.
- Group** Bus
- Syntax** `BUS:B<x>:SPI:DATA:SIZE <NR1>`
`BUS:B<x>:SPI:DATA:SIZE?`
- Arguments** *B<x>* is the number of the bus waveform.
- <NR1>* is the data size for the specified bus. The minimum value is 2 and maximum is 32.
- Examples** `BUS:B1:SPI:DATA:SIZE 7` sets the data size for SPI Bus 1 to seven bits per word.
- `BUS:B1:SPI:DATA:SIZE?` might return `BUS:B1:SPI:DATA:SIZE 8`, indicating that the data size for SPI Bus 1 is set to eight bits per word.

BUS:B<x>:SPI:DATA:SOURce

This command sets or queries the SPI Data (Data) source for the bus number specified by *x*.

- Conditions** Requires option 2-SERIAL or 2-ULTIMATE.
- Group** Bus
- Syntax** `BUS:B<x>:SPI:DATA:SOURce {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}`
`BUS:B<x>:SPI:DATA:SOURce?`

Arguments **B<x>** is the number of the bus waveform.

CH<x> designates an analog channel as the data source for the specified SPI bus.

DCH<x>_D<x> specifies the digital channel as the bus SPI clock source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> designates a math waveform as the data source.

REF<x> designates a reference waveform as the data source.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified SPI bus.

Examples **BUS:B1:SPI:DATA:SOURCE CH2** sets the SPI data source for the Bus 1 to CH2.

BUS:B3:SPI:DATA:SOURCE? might return **BUS:B3:SPI:DATA:SOURCE CH3**, indicating that the SPI data source for the Bus 3 is set to CH3.

BUS:B<x>:SPI:DATA:THReshold

This command sets or queries the SPI Data (Data) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax **BUS:B<x>:SPI:DATA:THReshold <NR3>**
BUS:B<x>:SPI:DATA:THReshold?

Arguments **B<x>** is the number of the bus waveform.

<NR3> is the SPI Data (SDA) source threshold for the specified bus.

Examples **BUS:B1:SPI:DATA:THReshold 50.0e-3** sets the threshold to 50.0 mV.

BUS:B1:SPI:DATA:THReshold? might return
BUS:B1:SPI:DATA:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:FRAMING

This command sets or queries the SPI framing setting for the specified bus. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus
Syntax	BUS:B<x>:SPI:FRAMING {IDLE SS} BUS:B<x>:SPI:FRAMING?
Arguments	B<x> is the number of the bus waveform. IDLE specifies IDLE SPI framing. SS specifies SS SPI framing.
Examples	BUS:B1:SPI:FRAMING IDLE sets the SPI framing to IDLE. BUS:B1:SPI:FRAMING? might return BUS:B1:SPI:FRAMING SS, indicating that the SPI framing is set to SS.

BUS:B<x>:SPI:IDLETime

This command sets or queries the SPI idle time for the specified bus. The bus is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Bus
Syntax	BUS:B<x>:SPI:IDLETime <NR3> BUS:B<x>:SPI:IDLETime?
Arguments	B<x> is the number of the bus waveform. <NR3> specifies the SPI idle time.
Examples	BUS:B1:SPI:IDLETime 0.000004 sets the idle time to 4 μ s. BUS:B1:SPI:IDLETime? might return BUS:B1:SPI:IDLETIME 5.0000E-6, indicating that the idle time is set to 5 μ s.

BUS:B<x>:SPI:MISo:DATA:POLarity

This command sets or queries the SPI MISo Data source polarity for the specified bus. The bus is specified by x.

Conditions	Requires option SR-SPI.
Group	Bus
Syntax	<code>BUS:B<x>:SPI:MISo:DATA:POLarity {HIGH LOW}</code> <code>BUS:B<x>:SPI:MISo:DATA:POLarity?</code>
Arguments	<code>B<x></code> is the number of the bus waveform. HIGH sets the polarity to high. LOW sets the polarity to low.
Examples	<code>BUS:B1:SPI:MISo:DATA:POLarity HIGH</code> sets the SPI MISo data polarity to high. <code>BUS:B1:SPI:MISo:DATA:POLarity?</code> might return <code>BUS:B1:SPI:MISo:DATA:POLarity HIGH</code> , indicating that the SPI MISo data polarity is high.

BUS:B<x>:SPI:MISo:INPut

This command sets or queries the SPI MISo source for the specified bus. The bus is specified by x.

Conditions	Requires option SR-SPI.
Group	Bus
Syntax	<code>BUS:B<x>:SPI:MISo:INPut {CH<x> DCH<x>_D<x> MATH<x> REF<x>}</code> <code>BUS:B<x>:SPI:MISo:INPut?</code>
Arguments	<code>B<x></code> is the number of the bus waveform. <code>CH<x></code> designates an analog channel as the source.

DCH<x>_D<x> specifies the channel bit of separate digital channel as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> designates a math waveform as the source.

REF<x> designates a reference waveform as the source.

Examples `BUS:B1:SPI:MISO:INPut Ch5` sets the source to channel 5.

`BUS:B1:SPI:MISO:INPut?` might return `BUS:B1:SPI:MISO:INPut HIGH`, indicating the source is set to channel 5.

BUS:B<x>:SPI:MISO:THReshold

This command sets or queries the SPI MISO Data source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax `BUS:B<x>:SPI:MISO:THReshold <NR3>`
`BUS:B<x>:SPI:MISO:THReshold?`

Arguments B<x> is the number of the bus waveform.
 <NR3> is the SPI MISO Data source threshold for the specified bus.

Examples `BUS:B1:SPI:MISO:THReshold 1.0` sets the Bus 1 SPI MISO threshold to 1.0 V.

`BUS:B1:SPI:MISO:THReshold?` might return
`:BUS:B1:SPI:MISO:THReshold 225.00000E-3`, indicates the Bus 1 SPI MISO threshold is set to 225.0 mV.

BUS:B<x>:SPI:MOSi:DATA:POLarity

This command sets or queries the SPI MOSI source polarity for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group	Bus
Syntax	BUS:B<x>:SPI:MOSi:DATA:POLarity {HIGH LOW} BUS:B<x>:SPI:MOSi:DATA:POLarity?
Arguments	B<x> is the number of the bus waveform. HIGH sets the polarity to high. LOW sets the polarity to low.
Examples	BUS:B1:SPI:MOSi:DATA:POLarity HIGH sets the SPI MOSI data polarity to high. BUS:B1:SPI:MOSi:DATA:POLarity? might return :BUS:B1:SPI:MOSi:DATA:POLarity HIGH, indicating that the SPI MOSI data polarity is high.

BUS:B<x>:SPI:MOSi:INPut

This command sets or queries the SPI MOSI source for the specified bus. The bus is specified by x.

Conditions	Requires option SR-SPI.
Group	Bus
Syntax	BUS:B<x>:SPI:MOSi:INPut {CH<x> DCH<x>_D<x> MATH<x> REF<x>} BUS:B<x>:SPI:MOSi:INPut?
Arguments	B<x> is the number of the bus waveform. CH<x> designates an analog channel as the source. DCH<x>_D<x> designates the channel bit of separate digital channel as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15. MATH<x> designates a math waveform as the source. REF<x> designates a reference waveform as the source.

Examples `BUS:B1:SPI:MOSi:INPut Ch5` sets the source to channel 5.
`BUS:B1:SPI:MOSi:INPut?` might return `:BUS:B1:SPI:MOSi:INPut HIGH`, indicating the source is set to channel 5.

BUS:B<x>:SPI:MOSi:THReshold

This command sets or queries the SPI MOSI source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax `BUS:B<x>:SPI:MOSi:THReshold <NR3>`
`BUS:B<x>:SPI:MOSi:THReshold?`

Arguments `B<x>` is the number of the bus waveform.
`<NR3>` is the SPI MOSI Data source threshold for the specified bus.

Examples `BUS:B1:SPI:MOSi:THReshold 1.0` sets the Bus 1 SPI MOSI threshold to 1.0 V.
`BUS:B1:SPI:MOSi:THReshold?` might return `:BUS:B1:SPI:MOSi:THReshold 225.00000E-3`, indicates the Bus 1 SPI MOSI threshold is set to 225.0 mV.

BUS:B<x>:SPI:NUMBer:INputs

This command sets or queries the number of inputs for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax `BUS:B<x>:SPI:NUMBer:INputs {ONE|TWO}`
`BUS:B<x>:SPI:NUMBer:INputs?`

Arguments B<x> is the number of the bus waveform.
ONE sets the Data Inputs as one.
TWO sets the Data Inputs as two.

Examples BUS:B1:SPI:NUMBER:INPUTS TWO sets the Data Inputs as two.
BUS:B1:SPI:NUMBER:INPUTS? might return :BUS:B1:SPI:NUMBER:INPUTS
TWO, indicating that the Data Inputs is two.

BUS:B<x>:SPI:SElect:POLarity

This command sets or queries the SPI Slave Select (SS) polarity for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax BUS:B<x>:SPI:SElect:POLarity {LOW|HIGH}
BUS:B<x>:SPI:SElect:POLarity?

Related Commands [BUS:B<x>:SPI:SElect:SOUrce](#)

Arguments B<x> is the number of the bus waveform.
LOW sets an active low polarity.
HIGH sets an active high polarity.

Examples BUS:B1:SPI:SELECT:POLARITY HIGH sets the SPI Slave Select polarity for the bus B1 to active high.
BUS:B1:SPI:SELECT:POLARITY? might return
BUS:B1:SPI:SELECT:POLARITY HIGH, indicating that the SPI Slave Select polarity for the bus B1 is set to active high.

BUS:B<x>:SPI:SElect:SOUrce

This command sets or queries the SPI Slave Select (SS) source for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax `BUS:B<x>:SPI:SElect:SOURCE {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}`
`BUS:B<x>:SPI:SElect:SOURCE?`

Related Commands [BUS:B<x>:SPI:SElect:POLarity](#)

Arguments B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the bus SPI Slave Select source.

DCH<x>_D<x> designates a digital channel as the bus SPI Slave Select source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> designates a math waveform as the Slave Select source.

REF<x> designates a reference waveform as the Slave Select source.

REF<x> _D<x> specifies a digital reference waveform as the clock source waveform for the specified SPI bus.

Examples `BUS:B1:SPI:SELECT:SOURCE MATH3` sets the SPI Slave Select source for the bus B1 to MATH3.

`BUS:B1:SPI:SELECT:SOURCE?` might return `BUS:B1:SPI:SELECT:SOURCE CH3`, indicating that the SPI Slave Select source for the bus B1 is set to CH3.

BUS:B<x>:SPI:SElect:THReshold

This command sets or queries the SPI Select (SS) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Bus

Syntax `BUS:B<x>:SPI:SElect:THReshold <NR3>`
`BUS:B<x>:SPI:SElect:THReshold?`

Arguments **B<x>** is the number of the bus waveform.
 <NR3> is the SPI Select (SS) source threshold for the specified bus.

Examples **BUS:B1:SPI:SElect:THResho1d 50.0e-3** sets the threshold to 50.0 mV.
BUS:B1:SPI:SElect:THResho1d? might return
BUS:B1:SPI:SELECT:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:TYPE

This command sets or queries the bus type or standard for the specified bus. The bus is specified by x. Arguments for a bus type are only available then the required serial bus option is installed.

Group Bus

Syntax **BUS:B<x>:TYPE {CAN| I2C| LIN| PARAllel| RS232C| SENT| SPI}**
BUS:B<x>:TYPE?

Arguments **B<x>** is the number of the bus waveform.
 CAN specifies a Controller Area Network bus.
 I2C specifies the Inter-IC bus.
 LIN specifies a Local Interconnect Network bus.
 PARAllel specifies a parallel bus.
 RS232C specifies the RS-232 Serial bus.
 SENT specifies the Single Edge Nibble Transmission (SENT) automotive serial bus.
 SPI specifies the Serial Peripheral Interface bus.

Examples **BUS:B1:TYPE I2C** sets the bus B1 type to I2C.
BUS:B1:TYPE? might return **BUS:B1:TYPE SPI**, indicating that the bus B1 type is set to SPI and that the bus behavior is governed by the **BUS:B1:SPI:xxxx** commands.

BUS:DELeTe (No Query Form)

This command deletes the specified bus.

Group	Bus
Syntax	BUS:DELeTe <QString>
Arguments	<QString> specifies the bus to delete and is of the form "B<NR1>", where <NR1> is ≥ 1 .
Examples	BUS:DELETE B1 deletes bus B1.

BUS:LIST? (Query Only)

This query returns a comma separated list of all currently defined buses.

Group	Bus
Syntax	BUS:LIST?
Returns	Returns all currently defined buses.
Examples	BUS:LIST? might return BUS:LIST B1,B4 indicating the bus 1 and bus 4 are defined.

BUSTABLE:ADDNew (No Query Form)

Adds the specified bus table. Argument is of the form "TABLE<NR1>", where <NR1> is ≥ 1).

Group	Bus
Syntax	BUSTABLE:ADDNew <QString>
Arguments	<QString> is a quoted string that is the name of the new bus table.

Examples `BUSTABLE:ADDNEW "Table1"` adds bus table Table1.

BUSTABLE:DELEte (No Query Form)

Deletes the specified bus table. Argument is of the form "TABLE<NR1>", where <NR1> is ≥ 1).

Group Bus

Syntax `BUSTABLE:DELEte <QString>`

Arguments <QString> is a quoted string that is the name of the bus table to delete.

Examples `BUSTABLE:DELETE "Table1"` deletes bus table Table1.

BUSTABLE:LIST? (Query Only)

This query lists all currently defined bus tables.

Group Bus

Syntax `BUSTABLE:LIST?`

Returns Returns a list of all currently defined bus tables.

Examples `BUSTABLE:LIST?` might return `BUSTABLE:LIST TABLE1` indicating TABLE1 is currently the only defined bus table.

BUSY? (Query Only)

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

Group Status and Error

Syntax BUSY?

Related Commands *OPC
 *WAI

Returns <NR1> = 0 means that the instrument is not busy processing a command whose execution time is extensive.
 <NR1> = 1 means that the instrument is busy processing Commands that Generate an OPC Message(See Table 3-3.).

Examples BUSY? might return BUSY 1, indicating that the instrument is currently busy.

*CAL? (Query Only)

This query-only command starts signal path calibration (SPC) and returns the status upon completion.

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 20 minutes on an 4-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Group Calibration

Syntax *CAL?

Returns 0 indicates SPC passed.
 -1 indicates SPC failed or did not complete.

Examples *CAL? starts the signal path calibration and returns the status upon completion.

CALibrate? (Query Only)

This query returns the status of signal path calibration.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes.

Group	Calibration
Syntax	CALibrate?
Examples	CALIBRATE? might return CALIBRATE:INTERNAL:STATUS PASS, indicating the calibration status.

CALibrate:INTERNAL (No Query Form)

This command (no query form) starts the signal path calibration (SPC) of the instrument. You can use the [CALibrate:INTERNAL:STATUS?](#) query to return the current status of the signal path calibration of the instrument.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 20 minutes on an 4-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group	Calibration
Syntax	CALibrate:INTERNAL
Related Commands	CALibrate:INTERNAL:STATUS?
Examples	CALIBRATE:INTERNAL starts the signal path calibration of the instrument.

CALibrate:INTERNAL:START (No Query Form)

This command (no query form) starts the signal path calibration (SPC) of the analog channels. This command is the same as the [CALibrate:INTERNAL](#) command. You can use the [CALibrate:INTERNAL:STATUS?](#) query to return the current status of the signal path calibration of the instrument.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 20 minutes on an 4-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group Calibration

Syntax CALibrate:INTERNAL:START

Related Commands [CALibrate:INTERNAL:STATUS?](#)

Examples CALIBRATE:INTERNAL:START starts the signal path calibration.

CALibrate:INTERNAL:STATUS? (Query Only)

This query-only command returns the current status of the signal path calibration.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 15 minutes per channel which means a total of 2 hours on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group Calibration

Syntax CALibrate:INTERNAL:STATUS?

Related Commands [*CAL?](#)

Returns This query will return one of the following:

- INIT indicates the instrument has not had signal path calibration run. The instrument may need to be readjusted at the Tektronix service center.
- PASS indicates that the signal path calibration completed successfully.
- FAIL indicates that the signal path calibration did not complete successfully.

Examples CALIBRATE:INTERNAL:STATUS? might return PASS, indicating that the current status of the signal path calibration is that the signal path calibration completed successfully.

CALibrate:PWRUpstatus? (Query Only)

This query-only command returns the current status of the power-up calibration.

Group Calibration

Syntax CALibrate:PWRUpstatus?

Returns This query will return one of the following:

- 0 Indicating the power-up calibration failed.
- 1 Indicating the power-up calibration passed.

Examples CALIBRATE:PWRUPSTATUS? might return 0, indicating that the power-up calibration failed.

CALLOUTS:CALLOUT<x> (No Query Form)

This command creates a new note. A Note is the default callout value. The callout number is specified by x.

Group Callouts

Syntax CALLOUTS:CALLOUT<x>

Examples CALLOUTS:CALLOUT1 creates a new Note callout specified as callout1.

CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE

This command sets or queries the source of the Bookmark callout type.

Group Callouts

Syntax	<code>CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE {CH1 CH2 CH3 CH4}</code> <code>CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE?</code>
Arguments	CH1 specifies the bookmark callout source as Ch1. CH2 specifies the bookmark callout source as Ch2. CH3 specifies the bookmark callout source as Ch3. CH4 specifies the bookmark callout source as Ch4.
Examples	<code>CALLOUTS:CALLOUT1:BOOKMark:SOURCE CH2</code> sets the bookmark callout source as Ch2 for callout1. <code>CALLOUTS:CALLOUT1:BOOKMark:SOURCE?</code> , might return "CH1", indicating that the bookmark callout source is Ch1.

CALLOUTS:CALLOUT<x>:BOOKMark:XPOS

This command sets or queries the X-Position of the Bookmark callout type.

Group	Callouts
Syntax	<code>CALLOUTS:CALLOUT<x>:BOOKMark:XPOS <NR1></code> <code>CALLOUTS:CALLOUT<x>:BOOKMark:XPOS?</code>
Arguments	<NR1> specifies the location of the bookmark linked to the source waveform in X-axis.
Examples	<code>CALLOUTS:CALLOUT1:BOOKMark:XPOS 9.28µs</code> sets the bookmark callout X-Position to 9.28 µs for callout1. <code>CALLOUTS:CALLOUT1:BOOKMark:XPOS?</code> , might return "9.28 µs", indicating the bookmark callout X-Position is 9.28 µs.

CALLOUTS:CALLOUT<x>:COLOR

This command sets or queries the text color of the callout.

Group	Callouts
--------------	----------

Syntax `CALLOUTS:CALLOUT<x>:COLOR <QString>`
`CALLOUTS:CALLOUT<x>:COLOR?`

Arguments `<QString>` specifies the callout text color using hexadecimal color values.

Examples `CALLOUTS:CALLOUT1:COLOR "#FF0000"` sets the text color of callout 1 to red.
`CALLOUTS:CALLOUT1:COLOR?`, might return `"#FF0000"`, indicating the text color is red.

CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:X

This command sets or queries horizontal display position of the callout text.

Group Callouts

Syntax `CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:X <NR1>`
`CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:X?`

Arguments `<NR1>` specifies the callout horizontal display position.

Examples `CALLOUTS:CALLOUT1:DISPLAYPOSITION:X 50` sets the callout text in the center on the horizontal axis of the display on the scopeapp.
`CALLOUTS:CALLOUT1:DISPLAYPOSITION:X?`, might return `"50"`, indicating the callout text is in the center on the horizontal axis of the display on the scopeapp

CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:Y

This command sets or queries vertical display position of the callout text.

Group Callouts

Syntax `CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:Y <NR1>`
`CALLOUTS:CALLOUT<x>:DISPLAYPOSITION:Y?`

Arguments `<NR1>` specifies the callout vertical display position.

Examples `CALLOUTS:CALLOUT1:DISPLAYPOSITION:Y 50` sets the callout text in the center on the vertical axis of the display on the scopeapp.

`CALLOUTS:CALLOUT1:DISPLAYPOSITION:Y?`, might return "50", indicating the callout text is in the center on the vertical axis of the display on the scopeapp

CALLOUTS:CALLOUT<x>:FONT:BOLD

This command sets or queries the bold state of the callout text.

Group Callouts

Syntax `CALLOUTS:CALLOUT<x>:FONT:BOLD {1|0}`
`CALLOUTS:CALLOUT<x>:FONT:BOLD?`

Arguments 1 specifies the callout font weight as bold.
 0 specifies the callout font weight as normal.

Examples `CALLOUTS:CALLOUT1:FONT:BOLD 1` sets the font weight of callout1 to bold.

`CALLOUTS:CALLOUT1:FONT:BOLD?`, might return "0", indicating the font weight is set to normal.

CALLOUTS:CALLOUT<x>:FONT:ITALIC

This command sets or queries the italic state of the callout text.

Group Callouts

Syntax `CALLOUTS:CALLOUT<x>:FONT:ITALIC {1|0}`
`CALLOUTS:CALLOUT<x>:FONT:ITALIC?`

Arguments 1 specifies the callout font style as italic.
 0 does not specify the font style as italic.

Examples `CALLOUTS:CALLOUT1:FONT:ITALIC 1` sets the font style of callout1 to italic.

`CALLOUTS:CALLOUT1:FONT:ITALIC?`, might return "0", indicating the font style is not italic.

CALLOUTS:CALLOUT<x>:FONT:SIZE

This command sets or queries the font size of the callout text.

Group	Callouts
Syntax	CALLOUTS:CALLOUT<x>:FONT:SIZE <NR1> CALLOUTS:CALLOUT<x>:FONT:SIZE?
Arguments	<NR1> specifies the font size in points.
Examples	CALLOUTS:CALLOUT1:FONT:SIZE 11 sets the font size of callout1 to 11 points. CALLOUTS:CALLOUT1:FONT:SIZE?, might return "14", indicating the font size is 14 points.

CALLOUTS:CALLOUT<x>:FONT:TYPE

This command sets or queries type of the callout.

Group	Callouts
Syntax	CALLOUTS:CALLOUT<x>:FONT:TYPE <QString> CALLOUTS:CALLOUT<x>:FONT:TYPE?
Arguments	<QString> specifies the type of font for the callout text. The available font types include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.
Examples	CALLOUTS:CALLOUT1:FONT:TYPE "Serif" sets the font type of callout1 to serif. CALLOUTS:CALLOUT1:FONT:TYPE?, might return "Monospace", indicating the font type is monospace.

CALLOUTS:CALLOUT<x>:FONT:UNDERLINE

This command sets or queries the underline state of the callout text.

Group	Callouts
--------------	----------

Syntax	CALLOUTS:CALLOUT<x>:FONT:UNDERLine {1 0} CALLOUTS:CALLOUT<x>:FONT:UNDERLine?
Arguments	1 underlines the callout text. 0 does not underline the callout text.
Examples	CALLOUTS:CALLOUT1:FONT:UNDERLine 1 displays the text of callout 1 as underlined. CALLOUTS:CALLOUT1:FONT:UNDERLine?, might return "0", indicating the callout text is not underlined.

CALLOUTS:CALLOUT<x>:TEXT

This command sets or queries the callout text.

Group	Callouts
Syntax	CALLOUTS:CALLOUT<x>:TEXT <QString> CALLOUTS:CALLOUT<x>:TEXT?
Arguments	<QString> specifies the callout text.
Examples	CALLOUTS:CALLOUT1:TEXT "Review measurements" sets the text for callout 1 as "Review measurements". CALLOUTS:CALLOUT1:TEXT?, might return "Review measurements", indicating the text for callout 1 is "Review measurements".

CALLOUTS:CALLOUT<x>:TYPE

This command sets or queries type of the callout.

Group	Callouts
Syntax	CALLOUTS:CALLOUT<x>:TYPE {NOTE ARROW RECTANGLE BOOKMARK} CALLOUTS:CALLOUT<x>:TYPE?

Arguments	NOTE specifies callout type as note. ARROW specifies callout type as arrow. RECTANGLE specifies callout type as rectangle. BOOKMARK specifies callout type as bookmark.
Examples	CALLOUTS:CALLOUT1:TYPE NOTE sets the callout type of callout1 to note. CALLOUTS:CALLOUT1:TYPE?, might return "ARROW", indicating the callout type is arrow.

CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x.

Group	Vertical
Syntax	CH<x>?
Examples	CH1? might return the following vertical parameters for channel 1: CH1:BANDWIDTH 1.0000E+09;COUPLING DC;DESKEW0.0000E+00;OFFSET 0.0000E+00;POSITION 0.0000E+00;SCALE 5.0000E-01;RESISTANCE 1.0000E+06;UNITS "V";ID:TYPE "1X";SERNUMBER "N/A";:CH1:PROBEFUNC:EXTATTEN 1.0000E+00;EXTUNITS "None";:CH1:LABEL:NAME "";XPOS 5;YPOS 5.

CH<x>:BANDwidth

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x.

The query form of this command always returns the approximate realized bandwidth of the channel.

Available arguments depend upon the instrument and the attached accessories.

Group	Vertical
Syntax	CH<x>:BANDwidth {<NR3> FUL1} CH<x>:BANDwidth?

Arguments CH<x> is the channel number.

<NR3> is the desired bandwidth. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value to set the upper bandwidth.

FULL disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.

Examples CH1:BANDWIDTH 20 sets the bandwidth of Channel 1 to 20 MHz.

CH2:BANDWIDTH?, might return :CH2:BANDWIDTH 500.0000E+06, indicating that there is bandwidth limiting on Channel 2.

CH<x>:CLIPping? (Query Only)

Queries whether the specified channel's input signal is clipping (exceeding) the channel A/D converter range. The channel is specified by x.

Group Vertical

Syntax CH<x>:CLIPping?

Related Commands [CH<x>:SCALE](#)
[CH<x>:POSITION](#)

Returns This query will return one of the following:

0 indicates the channel is not clipping.

1 indicates the channel is clipping.

Examples CH4:CLIPping? might return CH4:CLIPping? 1, indicating that the input signal range is exceeding (clipping) the channel 4 A/D converter range.

CH<x>:COUPLing

This command sets or queries the input coupling setting for the specified analog channel. The channel is specified by x.

NOTE. *The available arguments depend on the attached accessories.*

Group	Vertical
Syntax	CH<x>:COUPLing {AC DC DCREJ} CH<x>:COUPLing?
Arguments	CH<x> is the channel number. AC sets the specified channel to AC coupling. DC sets the specified channel to DC coupling. DCREJect sets DC Reject coupling when probes are attached that support that feature.
Examples	CH2:COUPLING AC sets Channel 2 coupling to AC. CH3:COUPLING? might return :CH3:COUPLING DC, indicating that Channel 3 is set to DC coupling.

CH<x>:DESKew

This command sets or queries the horizontal deskew time for the specified channel. The channel is specified by x.

Group	Vertical
Syntax	CH<x>:DESKew <NR3> CH<x>:DESKew?
Arguments	CH<x> is the channel number. <NR3> is the deskew time for this channel, ranging from -125 ns to +125 ns with a resolution of 40 ps. Out-of-range values are clipped.
Examples	CH4:DESKEW 5.0E-9 sets the deskew time for Channel 4 to 5 ns. CH2:DESKEW? might return CH2:DESKEW 2.0000E-09, indicating that the deskew time for Channel 2 is set to 2 ns.

CH<x>:DITHerRange

This command sets or returns the amount of dithering for the specified analog channel. The channel is specified by x. The amount of dithering is a percentage of full scale (10 times volts/division).

Note: Setting this value to 0.0 for any unused channels may slightly improve performance.

Group	Vertical
Syntax	CH<x>:DITHerRange <NR3>
Arguments	<NR3> is the amount of dithering as a percentage of full scale. Must be between 0.0 and 100.0 and 0.0 disables dithering.
Examples	<p>CH1:DITHerRange 10 sets the amount of dithering as a percentage of full scale to 10.</p> <p>CH1:DITHerRange? might return CH1:DITHerRange 10, which if vertical scale is 1V/div then the dithering amplitude will be 1 Vpk-pk.</p>

CH<x>:INVert

This command sets or queries invert state of the specified channel. The channel is specified by x.

Group	Vertical
Syntax	CH<x>:INVert {ON OFF <NR1>} CH<x>:INVert?
Arguments	<p>OFF turns off the channel invert.</p> <p>ON turns on the channel invert.</p> <p><NR1> 0 turns off the channel invert; any other value turns on the channel invert.</p>
Examples	<p>CH2:INVert OFF turns off the invert on channel 2.</p> <p>CH2:INVert? might return CH2:INVert 1, indicating channel 2 is inverted.</p>

CH<x>:LABel:COLor

This command sets or queries the color of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:COLor <QString>

Arguments CH<x> is the channel number.
<QString> is the label color. To return the color to the default color, send an empty string as in this example: CH5:LABel:COLOR "".

Examples CH2:LABel:COLor "#FFFF00" sets the label color to yellow.
CH2:LABel:COLor? might return CH2:LABel:COLOR "#FF0000" indicating the color is red.

CH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.
OFF argument turns off bold font.
ON argument turns on bold font.
<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples CH2:LABel:FONT:BOLD OFF turns off the bold font.
CH2:LABel:FONT:BOLD? might return CH2:LABel:FONT:BOLD 1 indicating a bold font.

CH<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.
 OFF argument turns off italic font.
 ON argument turns on italic font.
 <NR1> = 0 turns off italic font; any other value turns on italic font.

Examples CH2:LABel:FONT:ITALic ON set the font to italic.
 CH2:LABel:FONT:ITALic? might return CH2:LABEL:FONT:ITALIC 0 indicating the font is not italic.

CH<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:SIZE <NR1>

Arguments CH<x> is the channel number.
 <NR1> is the font size.

Examples CH2:LABel:FONT:SIZE 16 sets the font size to 16 point size.
 CH2:LABel:FONT:SIZE? might return CH2:LABEL:FONT:SIZE 20 indicating a 20 point font size.

CH<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified channel label, such as Arial or Times New Roman. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:TYPE <QString>

Arguments CH<x> is the channel number.
<QString> is the specified font type.

Examples CH2:LABel:FONT:TYPE "Monospace" sets the font to a mono space font.
CH2:LABel:FONT:TYPE? might return CH2:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

CH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.
OFF argument turns off underlined font.
ON argument turns on underlined font.
<NR1> = 0 turns off underlined font; any other value turns on underlined font.

Examples CH2:LABel:FONT:UNDERline ON sets the font to underlined.
CH2:LABel:FONT:UNDERline? might return CH2:LABEL:FONT:UNDERLINE 0 indicating the font is not underlined.

CH<x>:LABel:NAME

This command sets or queries the label attached to the displayed waveform for the specified channel. The channel is specified by x.

Group	Vertical
Syntax	CH<x>:LABel:NAME <QString> CH<x>:LABel:NAME?
Arguments	CH<x> is the channel number. <QString> is an alphanumeric character string, ranging from 1 through 32 characters in length.
Examples	CH2:LABel:NAME "Pressure" changes the waveform label for the Channel 2 waveform to "Pressure". CH3:LABel:NAME? might return CH3:LABel:NAME "Force", indicating that the waveform label for the Channel 3 waveform is "Force".

CH<x>:LABel:XPOS

This command sets or queries the X-position of the specified channel label. The channel is specified by x.

Group	Vertical
Syntax	CH<x>:LABel:XPOS <NR3> CH<x>:LABel:XPOS?
Arguments	CH<x> is the channel number. <NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen.
Examples	CH3:LABel:XPOS 5 moves the waveform label for Channel 3 so that it begins 5 pixels to the right of the left edge of the screen. CH2:LABel:XPOS? might return CH2:LABel:XPOS 5, indicating that the waveform label for the Channel 2 currently 5 pixels to the right of the left edge of the screen.

CH<x>:LABel:YPOS

This command sets or queries the Y-position of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:YPOS <NR3>
CH<x>:LABel:YPOS?

Arguments CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the baseline of the waveform. Positive values are above the baseline and negative values are below.

Examples CH3:LABel:YPOS -20 moves the waveform label for the Channel 3 20 pixels below the baseline of the waveform.

CH2:LABel:YPOS? might return CH2:LABel:YPOS 0, indicating that the waveform label for the Channel 2 is currently located at the baseline of the waveform.

CH<x>:OFFSet

This command sets or queries the vertical offset for the specified analog channel.

Group Vertical

Syntax CH<x>:OFFSet <NR3>
CH<x>:OFFSet?

Arguments CH<x> is the channel number.

<NR3> is the offset value for the specified channel.

Examples CH3:OFFSet 2.0E-3 sets the offset for Channel 3 to 2 mV.

CH4:OFFSet? might return :CH4:OFFSet 1.0000E-03, indicating that the offset for Channel 4 is set to 1 mV.

CH<x>:POSition

This command sets or queries the vertical position for the specified analog channel.

Group Vertical

Syntax CH<x>:POSition <NR1>

Arguments CH<x> is the channel number.
<NR1> is the vertical position for the specified analog channel.

Examples CH2:POSition -2.0 sets the position to -2 divisions.
CH2:POSition? might return CH2:POSITION -2.2400 indicating the position is -2.24 divisions.

CH<x>:PROBEFunc:EXTAtten

This command is used to specify the attenuation value as a multiplier to the given scale factor on the specified channel. The channel is specified by x.

The query form of this command returns the user-specified attenuation.

Group Vertical

Syntax CH<x>:PROBEFunc:EXTAtten <NR3>
CH<x>:PROBEFunc:EXTAtten?

Related Commands [CH<x>:PROBEFunc:EXTDBatten](#)

Arguments CH<x> is the channel number.
<NR3> is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

Examples CH1:PROBEFUNC:EXTATTEN 167.00E-3 specifies an external attenuation, which is connected between the your input signal and the input of the probe attached to Channel 1.

CH2:PROBEFUNC:EXTATTEN? might return CH2:PROBEFUNC:EXTATTEN 1.0000E+00, indicating that the probe attached to Channel 2 is connected directly to the user's signal.

CH<x>:PROBEFunc:EXTDBatten

This command sets or queries the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument input channels. The channel is specified by x.

The query form of this command returns the user-specified attenuation in decibels.

Group Vertical

Syntax CH<x>:PROBEFunc:EXTDBatten <NR3>
CH<x>:PROBEFunc:EXTDBatten?

Related Commands [CH<x>:PROBEFunc:EXTAtten](#)

Arguments CH<x> is the channel number.

<NR3> is the attenuation value, which is specified in the range from -200.00 dB to 200.00 dB.

Examples CH3:PROBEFUNC:EXTDBATTEN 2.5 specifies an external 2.5 dB attenuator on Channel 3.

CH1:PROBEFUNC:EXTDBATTEN? might return CH1:PROBEFUNC:EXTDBATTEEN 2.5000E+00, indicating that the attenuation for Channel 1 is 2.5 dB.

CH<x>:PROBEFunc:EXTUnits

This command Sets or queries the units of the specified channel. The channel is specified by x. This command can only be set to "V" or "A", which corresponds to selecting a voltage or current probe respectively. The [CH<x>:PROBEFunc:EXTUnits](#) command can only select "V" if the [CH<x>:PROBEFunc:EXTUnits:STATE](#) is disabled (set to 0).

Group Vertical

Syntax CH<x>:PROBEFunc:EXTUnits <QString>
CH<x>:PROBEFunc:EXTUnits?

Arguments	CH<x> is the channel number. <QString> a quoted string that indicates the units of the specified channel. Only units "V" or "A" are supported. This command is case sensitive.
Examples	CH4:PROBEFunc:EXTUNITS "V" sets the unit of measurement for the Channel 4 to Volts. CH2:PROBEFunc:EXTUNITS? might return :CH2:PROBEFunc:EXTUNITS "V", indicating that the unit of measurement for Channel 2 is Volts.

CH<x>:PROBEFunc:EXTUnits:STATE

This command sets or queries measure current status as ON or OFF. The channel is specified by x. If this command is set, the vertical scale is set to "A", as it implies that the unit is measuring current from a voltage probe. When it is unset, the vertical scale is set to the value of [CH<x>:PROBEFunc:EXTUnits](#) ("V" or "A").

Group	Vertical
Syntax	CH<x>:PROBEFunc:EXTUnits:STATE {ON OFF <NR1>}
Arguments	CH<x> is the channel number. OFF argument turns current status off. ON argument turns current status on. <NR1> = 0 turns current status off; any other value turns current status on.
Examples	CH2:PROBEFunc:EXTUnits:STATE ON sets the CH2 measure current status to on. CH2:PROBEFunc:EXTUnits:STATE? might return CH2:PROBEFunc:EXTUNITS:STATE 0 indicating that CH2 current status is off.

CH<x>:SCALE

This command sets or returns the vertical scale for the specified analog channel. The channel is specified by x.

Group	Vertical
--------------	----------

Syntax CH<x>:SCALE <NR3>

Arguments CH<x> is the channel number.
<NR3> is the vertical scale for the specified analog channel.

Examples CH2:SCALE 200E-3 sets the scale to 200 mV per division.
CH2:SCALE? might return CH2:SCALE 500.0000E-3 indicating the vertical scale for the specified channel is 500 mV per division.

CH<x>:SCALERATIO

This command sets or returns the scale ratio for the specified analog channel.

Group Vertical

Syntax CH<x>:SCALERATIO <NR2>
CH<x>:SCALERATIO?

Arguments CH<x> is the channel number.
<NR2> is the scale ratio for the specified analog channel.

Examples CH2:SCALERATIO 2.0 sets the scale ratio to 2.0.
CH2:SCALERATIO? might return CH2:SCALERATIO 1.0000 indicating the scale ratio is 1.0.

CH<x>:TERMINATION

This command sets or queries the vertical termination for the specified analog channel. The channel is specified by x.

NOTE. The available arguments depend on the instrument model and the attached accessories.

Group Vertical

Syntax	CH<x>:TERmination <NR3> CH<x>:TERmination?
Arguments	CH<x> is the channel number. <NR3> specifies the channel input resistance, which can be specified as 50 Ω or 1,000,000 Ω .
Examples	CH4:TERMINATION 50.0E+0 establishes 50 Ω impedance on Channel 1. CH2:TERMINATION? might return CH2:TERMINATION 50.0E+0, indicating that Channel 2 is set to 50 Ω impedance.

CH<x>:VTERM:BIAS

Sets or queries the termination bias voltage for the specified channel (if control is available).

Conditions	Requires a probe with dual inputs.
Group	Vertical
Syntax	CH<x>:VTERM:BIAS <NR3> CH<x>:VTERM:BIAS?
Arguments	CH<x> is the channel number. <NR> is the termination voltage.
Examples	CH1:VTERM:BIAS 1.5 sets the termination bias voltage on channel 1 to 1.5 Volts CH2:VTERM:BIAS? might return CH2:VTERM:BIAS 0.0000, indicating that the termination bias voltage for channel 2 is set to 0 volts.

CLEAR (No Query Form)

This command clears acquisitions, measurements, and waveforms.

Group	Miscellaneous
--------------	---------------

Syntax CLEAR

Examples CLEAR clears all acquisitions, measurements, and waveforms.

*CLS (No Query Form)

This command (no query form) clears the following:

- Event Queue
- Standard Event Status Register
- Status Byte Register (except the MAV bit)

If the *CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) control message will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV.

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a single sequence acquisition operation is still being processed when the *CLS command is executed.

Group Status and Error

Syntax *CLS

Related Commands [DESE](#)
[*ESE](#)
[*ESR?](#)
[EVENT?](#)
[EVMsg?](#)
[*SRE](#)
[*STB?](#)

Examples *CLS clears the instrument status data structures.

CONFIGuration:ANALOG:BandWidth? (Query Only)

This command queries the maximum licensed bandwidth of the instrument.

Group	Vertical
Syntax	CONFIGuration:ANALOG:BandWidth?
Returns	The maximum licensed bandwidth of the instrument is returned.
Examples	CONFIGuration:ANALOG:BandWidth? might return CONFIGURATION:ANALOG:BandWidth 2.0000E+9 indicating the bandwidth is 2.0 GHz.

CONNected:REQUested:STATus (No Query Form)

This command sets the status of the Connected Scope Preference feature. After issuing the status command, the `CONNected:SAVEPREferences` command should be set immediately to make the feature enabled. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group	Miscellaneous
Syntax	CONNected:REQUested:STATus <NR1>
Arguments	<NR1> enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the feature.
Examples	CONNected:REQUested:STATus 1 CONNected:SAVEPREferences enables and saves Connected Scope Preference feature.

CONNected:STATus? (Query Only)

This query command returns the status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group	Miscellaneous
--------------	---------------

Syntax	<code>CONNECTed:STATUS?</code>
Returns	This command returns either 1 or 0. A return value of 1 means the feature is enabled and 0 means the feature is disabled.
Examples	<code>CONNECTed:STATUS?</code> might return <code>:CONNECTed:STATUS 1</code> , indicating the Connected Scope Preference feature is enabled.

CONNECTed:USAGe:TRack:REQUeSTed:STATUs (No Query Form)

This command sets the tracking usage status of the Connected Scope Preference feature. After issuing the status command, the `CONNECTed:SAVEPREferences` command should be set immediately to make the feature enabled. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group	Miscellaneous
Syntax	<code>CONNECTed:USAGe:TRack:REQUeSTed:STATUs<NR1></code>
Arguments	<NR1> enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the tracking usage status of the Connected Scope Preference feature.
Examples	<code>CONNECTed:USAGe:TRack:REQUeSTed:STATUs 1</code> <code>CONNECTed:SAVEPREferences</code> enables and saves the tracking usage status of the Connected Scope Preference feature.

CONNECTed:USAGe:TRack:STATUs? (Query Only)

This query command returns the tracking usage status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group	Miscellaneous
Syntax	<code>CONNECTed:USAGe:TRack:STATUs?</code>

Returns This command returns either 1 or 0. A return value of 1 means the feature is enabled and 0 means the feature is disabled.

Examples `CONNECTed:USAGe:TRAck:STATUs?` might return `:CONNECTed:USAGe:TRAck:STATUs 1`, indicating the tracking usage status of the Connected Scope Preference feature is enabled.

CURVe? (Query Only)

This command transfers waveform data from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The `CURVe?` query transfers data from the instrument. The data source is specified by the `DATA:SOURce` command. The first and last data points are specified by the `DATA:STARt` and `DATA:STOP` commands.

For MATH sources, only 8-byte double precision floating point data is returned in `:CURVe?` queries.

Group Waveform Transfer

Syntax `CURVe?`

Related Commands [DATA:SOURce](#)
[DATA:STARt](#)
[DATA:STOP](#)
[SAVe:WAVEform](#)
[SAVEON:WAVEform:FILEFormat](#)
[WFMOutpre?](#)

Returns `CH<x>_SV_NORMal`, `CH<x>_SV_AVErAge`, `CH<x>_SV_MAXHOld`, `CH<x>_SV_MINHOld`: When `DATA:SOURce` is set to one of these traces, the data points represent the amplitude of the Spectrum View frequency domain traces in the current units, and are floating point values. The frequency domain trace data is returned as 8-byte floating point values.

`CH<x>_SV_AMPliTude`, `CH<x>_SV_FREQuency`, `CH<x>_SV_PHASe`: When `DATA:SOURce` is set to one of these traces, the data points are the Magnitude vs. Time, Frequency vs. Time, or Phase vs. Time representations of the RF

input signal (Spectrum View) time domain traces. The RF time domain traces are returned as floating point values.

CH<x>_FREQ_vs_time is the frequency in Hz.

CH<x>_PHASE_vs_time is the phase in degrees.

CH<x>_MAG_vs_time is the magnitude in Volts.

CH<x>_SV_BASEband_IQ is the Spectrum View RF baseband IQ traces. When DATA:SOURce is set to one of these traces, the data points represent the digitizing levels of the IQ data. The IQ data is formatted as pairs of integer values (i.e. <I>,<Q>,<I>,<Q>,...)

Examples CURVE? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return :CURVE 61,62,61,60,60,-59,-59,-58,-58,-59

NOTE. *Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.*

*It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.*

CURVEStream? (Query Only)

This query-only command continuously transfers waveform data from the instrument as it is acquired.

This command puts the instrument into a streaming data mode, allowing the controller to receive waveform records as fast as they are acquired. Use the [DATA:SOURce](#) command to specify the waveform sources. The command supports all the same data formatting options as the [CURVE?](#) command.

- Control of the instrument through the user interface or other external clients is not allowed while in streaming data mode.
- The GPIB controller must take the instrument out of this streaming data mode to terminate the query and allow other input sources to resume communication with the instrument. The following options are available to transition out of streaming data mode:
 - Send a device clear over the bus
 - Send another command or query to the instrument
- Turning the waveform screen display mode off (:DISPlay:WAVEform OFF) may increase waveform throughput during streaming mode.
- Using a data encoding of SRIBinary (DATA:ENCdg SRIBinary) may also increase the waveform throughput since that is the raw native data format of the oscilloscope.
- While in streaming data mode, two extreme conditions can occur.
 - If the waveform records are being acquired slowly (high resolution), configure the controller for a long time-out threshold, as the data is not sent out until each complete record is acquired.
 - If the waveform records are being acquired rapidly (low resolution), and the controller is not reading the data off the bus fast enough, the trigger rate is slowed to allow each waveform to be sent sequentially.

Group Waveform Transfer

Syntax CURVEStream?

Related Commands [CURVe?](#)
[DATA:SOUrce](#)
[DATA:STARt](#)
[DATA:STOP](#)
[DATA:ENCdg](#)
[DATA:WIDth](#)
[WFMOuPre?](#)

Returns Formatted waveform data, once per acquisition.

Examples CURVESTREAM? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return :CURVESTREAM 61,62,61,60,60,-59,-59,-58,-58,-59

NOTE. Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.

It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.

DATA

This command sets or queries the format and location of the waveform data that is transferred with the CURVe? command.

Group Waveform Transfer

Syntax DATA {INIT|SNAP}
DATA?

Related Commands CURVe?
DATA:START
DATA:STOP
DATA:ENCdg
SAVEON:WAVEform:FILEFormat
WFMOutpre:NR_Pt?

Arguments INIT initializes the waveform data parameters to their factory defaults except for DATA:STOP, which is set to the current acquisition record length.

SNAP Sets DATA:START and DATA:STOP to match the current waveform cursor positions of WAVEVIEW1 CURSOR1 if these waveform cursors are currently

on. If these waveform cursors are not on when the DATA SNAP command is sent, it is silently ignored and DATA:START and :STOP remain unchanged.

If either of the cursors is outside the record, DATA:START and :STOP are set to nearest values within the record. This will cause DATA:START and/or :STOP to snap to the beginning or end of the record whichever is nearest, and the following event is set: 500, "Execution warning", "One or both cursors outside of waveform record".

If there is no acquired waveform for the cursor source, then DATA:START and :STOP remain unchanged and the following event is set: 500, "Execution warning", "No acquired waveform record".

If DATA:START is greater than :STOP, the values of DATA:START and STOP are swapped and the following event is set: 530, "Data start > stop, Values swapped internally".

Examples DATA INIT initializes the waveform data parameters to their factory defaults.

DATA? might return DATA:ENCDG RIBINARY;SOURCE CH1;START 1;STOP 1000;FRAMESTART 1;FRAMESTOP 1000.

DATA:ENCdg

This command sets or queries the format of outgoing waveform data. This command is equivalent to setting [WFMOupre:ENCdg](#), [WFMOupre:BN_Fmt](#), and [WFMOupre:BYT_Or](#). Setting the DATA:ENGdg value causes the corresponding WFMOupre values to be updated and vice versa.

NOTE. Values are constrained (for outbound data) to the format of the data specified by DATA:SOURce.

Group Waveform Transfer

Syntax DATA:ENCdg {ASCIi| RIBinary| RPBinary| FPBinary| SRIBinary| SRPbinary| SFPbinary}
DATA:ENCdg?

Related Commands [WFMOupre:ENCdg](#)
[WFMOupre:BN_Fmt](#)
[WFMOupre:BYT_Or](#)

- Arguments**
- **ASCIi** specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN_Fmt and :BYT_Or are ignored.
 - **RIBinary** specifies the positive integer data-point representation, with the most significant byte transferred first.
 When :BYT_Nr is 1, the range from 0 through 255. When :BYT_Nr is 2, the range is from 0 to 65,535. When :BYT_Nr is 4, then the waveform being queried would return Fast Acquisition Pixmap data (if fast acq is turned on and data mode is set to pixmap).
 - **RPBinary** specifies the positive integer data-point representation, with the most significant byte transferred first.
 When :BYT_Nr is 1, the range from 0 through 255. When :BYT_Nr is 2, the range is from 0 to 65,535.
 - **FPBinary** specifies the floating point (width = 4) data.
 The range is from -3.4×10^{38} to 3.4×10^{38} . The center of the screen is 0. The upper limit is the top of the screen and the lower limit is the bottom of the screen.
 The **FPBinary** argument is only applicable to math waveforms or ref waveforms saved from math waveforms.
 - **SRIbinary** is the same as **RIBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.
 - **SRPbinary** is the same as **RPBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
 - **SFPbinary** specifies floating point data in IBM PC format. The **SFPbinary** argument only works on math waveforms or ref waveforms saved from math waveforms.

Table 2-47: DATA and WFMOutpre Parameter Settings

DATA:ENCdg Setting	WFMOutpre Settings			
	:ENCdg	:BN_Fmt	:BYT_Or	:BYT_NR
ASCIi	ASC	N/A	N/A	1,2,4
RIBinary	BIN	RI	MSB	1,2
RPBinary	ASC	RP	MSB	1,2
FPBinary	BIN	FP	MSB	4
SRIbinary	BIN	RI	LSB	1,2
SRPbinary	BIN	RP	LSB	1,2
SFPbinary	BIN	FP	LSB	4

Examples DATA:ENCDG RPBinary sets the data encoding format to be a positive integer data-point representation, where the most significant byte is transferred first.

DATA:ENCDG? might return DATA:ENCDG SRPBINARY for the format of the outgoing waveform data.

DATA:MODE

This command sets or queries the mode for waveform data sent to returned by CURVE?. When the data mode is set as VECTor then you get the waveform sampled data. The Data width is reset to 1 or 2 instead of 4.

Group Waveform Transfer

Syntax DATA:MODE {VECTor|PIXmap}
DATA:MODE ?

Arguments VECTor sets the mode for waveform data to vector.
PIXmap sets the mode for waveform data to pixmap.

Examples DATA:MODE VECTor sets the mode for waveform data to vector.
DATA:MODE ? might return :DATA:MODE VECTor, indicating that the mode for waveform data is set to vector.

DATA:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting WFMOutpre:RESample. Setting the DATA:RESample value causes the corresponding WFMOutpre value to be updated and vice versa.

Group Waveform Transfer

Syntax DATA:RESample <NR1>
DATA:RESample?

Related Commands [WFMOutpre:RESample](#)

Arguments <NR1> is the resampling rate. The default value is 1, which means every sample is returned. A value of 2 returns every other sample, while a value of 3 returns every third sample, and so on.

Examples DATA:RESAMPLE 1 sets the resampling to return all waveform data samples.
 DATA:RESAMPLE? might return :DATA:RESAMPLE 4, indicating that every fourth sample of the waveform data is returned.

DATA:SOURce

This command sets or queries the location of waveform data that is transferred from the instrument by the CURVe?? Query.

Group Waveform Transfer

Syntax DATA:SOURce {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>}
 DATA:SOURce?

Related Commands [CURVe?](#)
[DATA](#)

Arguments CH<x> selects the specified analog channel as the source.
 DCH<x>_D<x> specifies the digital channel as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15 and ALL (all digital bits).
 MATH<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 4.
 REF<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 8.

Examples DATA:SOURce CH1 specifies that the CH1 waveforms will be transferred in the next CURVe? query.
 DATA:SOURce? might return DATA:SOURce REF3, indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

DATA:SOURce:AVAILable? (Query Only)

This query returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries. This means that the waveforms have been acquired. If there are none, NONE is returned.

Group Waveform Transfer

Syntax DATA:SOURce:AVAILable?

Related Commands [CURVe?](#)

Returns Returns a list of source waveforms that are currently available for :CURVe? queries.

Examples DATA:SOURCE:AVAILABLE? might return DATA:SOURCE:AVAILABLE CH1, CH2, CH3, CH4, DCH1_DALL, DCH1_D0, DCH1_D1, DCH1_D2, DCH1_D3, DCH1_D4, DCH1_D5, DCH1_D6, DCH1_D7, DCH1_D8, DCH1_D9, DCH1_D10, DCH1_D11, DCH1_D12, DCH1_D13, DCH1_D14, DCH1_D15, MATH1 indicating that CH1, CH2, CH3, CH4, DCH1_DALL, DCH1_D0, DCH1_D1, DCH1_D2, DCH1_D3, DCH1_D4, DCH1_D5, DCH1_D6, DCH1_D7, DCH1_D8, DCH1_D9, DCH1_D10, DCH1_D11, DCH1_D12, DCH1_D13, DCH1_D14, DCH1_D15, and MATH1 are available.

DATA:START

This command sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument.

Group Waveform Transfer

Syntax DATA:START <NR1>
DATA:START?

Related Commands [CURVe?](#)
[DATA](#)
[DATA:STOP](#)
[WFMOutpre:NR_Pt?](#)

Arguments <NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATA:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATA:START and DATA:STOP are order independent. When DATA:STOP is greater than DATA:START, the values will be swapped internally for the CURVE? query.

Examples DATA:START 10 specifies that the waveform transfer will begin with data point 10.

DATA:START? might return DATA:START 214, indicating that data point 214 is the first waveform data point that will be transferred.

DATA:STOP

This command sets or queries the last data point that will be transferred when using the CURVe?? query. This command allows for the transfer of partial waveforms to the controller.

Changes to the record length value are not automatically reflected in the data:stop value. As record length is varied, the DATA:STOP value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not automatically and correctly reflect increases in record length if the distance from DATA:START to DATA:STOP stays smaller than the increased record length.

NOTE. When using the CURVe? command, DATA:STOP is ignored.

Group Waveform Transfer

Syntax DATA:STOP <NR1>
DATA:STOP?

Related Commands [CURVe?](#)
[DATA](#)
[DATA:START](#)
[WFMOutpre:NR_Pt?](#)

Arguments <NR1> is the last data point that will be transferred, which ranges from 1 to the record length. If <NR1> is greater than the record length, then data will be transferred up to the record length. If both DATA:START and DATA:STOP are greater than the record length, the last data point in the record is returned.

DATA:START and DATA:STOP are order independent. When DATA:STOP is less than DATA:START, the values will be swapped internally for the CURVE? query.

If you always want to transfer complete waveforms, set DATA:START to 1 and DATA:STOP to the maximum record length, or larger.

- Examples** DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.
- DATA:STOP? might return DATA:STOP 14900, indicating that 14900 is the last waveform data point that will be transferred.

DATA:WIDTH

This command specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVE? query. (This command is synonymous with WFMOutpre:BYT_Nr.)

Group Waveform Transfer

Syntax DATA:WIDTH <NR1>
DATA:WIDTH?

Related Commands [WFMOutpre:BYT_Nr](#)

Arguments <NR1> is an integer that indicates the number of bytes per point for the outgoing waveform data when queried using the CURVE? command. For analog channels, NR1 can be 1 or 2.

Examples DATA:WIDTH 1 sets the data width to 1 byte.

DATA:WIDTH? might return DATA:WIDTH 1 indicating the width, in bytes per point, for waveform data transferred by the CURVE? query is 1 byte.

DATE? (Query Only)

This command queries the date that the instrument displays.

Group Miscellaneous

Syntax DATE?

Related Commands [TIME](#)

Returns <QString> is a date in the form “yyyy-mm-dd” where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

Examples DATE? might return DATE 2000-01-24, indicating the current date is set to January 24, 2000.

DCH<x>:D<x>:THReshold

This command sets or queries the threshold level in volts for the specified digital channel. If the source channel doesn't exist, a hardware missing error event is set.

Group Digital

Syntax DCH<x>:D<x>:THReshold <NR3>
DCH<x>:D<x>:THReshold?

Arguments DCH<x> specifies the digital channel. The supported value is 1.

D<x> specifies the digital bits. The supported values are 0 to 15. 0 to 7 have common a threshold and 8 to 15 have a common threshold.

<NR3> specifies the threshold level in volts for the specified digital channel. The minimum value is -20 V and the maximum value is 30.0 V.

Examples DCH1:D1:THRESHOLD 30.0 sets the threshold to 30.0 V.

DCH1:D1:THRESHOLD? might return :DCH1:D1:THRESHOLD 0.0 indicating the threshold is set to 0.0 V.

DCH<x>_D<x>:LABel:COLor

This command sets or queries the color of the label of the specified digital bit.

Group Digital

Syntax DCH<x>_D<x>:LABel:COLor <QString>
DCH<x>_D<x>:LABel:COLor?

- Arguments** DCH<x> specifies the digital channel. The supported value is 1.
 D<x> specifies the digital bits. The supported values are 0 to 15.
 <QString> is the label color. To return the color to the default color, send an empty string as in this example: :DCH1_D1:LABEL:COLOR "".
- Examples** DCH1_D1:LABEL:COLOR "#FF0000" sets the color to red.
 DCH1_D1:LABEL:COLOR? might return :DCH1_D1:LABEL:COLOR "#FFFF00", indicating the color is yellow.

DCH<x>_D<x>:LABEL:FONT:BOLD

This command sets or queries the bold state of the label of the specified digital bit.

- Group** Digital
- Syntax** DCH<x>_D<x>:LABEL:FONT:BOLD {<NR1>|OFF|ON}
 DCH<x>_D<x>:LABEL:FONT:BOLD?
- Arguments** DCH<x> specifies the digital channel. The supported value is 1.
 D<x> specifies the digital bits. The supported values are 0 to 15.
 <NR1> = 0 turns off bold font; any other value turns on bold font.
 OFF turns off bold font.
 ON turns on bold font.
- Examples** DCH1_D1:LABEL:FONT:BOLD ON turns on bold font.
 DCH1_D1:LABEL:FONT:BOLD? might return :DCH1_D1:LABEL:FONT:BOLD 0, indicating the font is not bold.

DCH<x>_D<x>:LABEL:FONT:ITALIC

This command sets or queries the italic state of the label of the specified digital bit.

- Group** Digital
- Syntax** DCH<x>_D<x>:LABEL:FONT:ITALIC {<NR1>|OFF|ON}
 DCH<x>_D<x>:LABEL:FONT:ITALIC?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
D<x> specifies the digital bits. The supported values are 0 to 15.
<NR1> = 0 turns off italic font; any other value turns on italic font.
OFF turns off italic font.
ON turns on italic font.

Examples DCH1_D1:LABe1:FONT:ITALic ON turns on italic font.
DCH1_D1:LABe1:FONT:ITALic? might return
:DCH1_D1:LABe1:FONT:ITALic 0, indicating the font is not italic.

DCH<x>_D<x>:LABe1:FONT:SIZE

This command sets or queries the font size of the label of the specified digital bit.

Group Digital

Syntax DCH<x>_D<x>:LABe1:FONT:SIZE <NR1>
DCH<x>_D<x>:LABe1:FONT:SIZE?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
D<x> specifies the digital bits. The supported values are 0 to 15.
<NR1> specifies the font size in points.

Examples DCH1_D1:LABe1:FONT:SIZE 16 sets the font size to 16 points.
DCH1_D1:LABe1:FONT:SIZE? might return :DCH1_D1:LABe1:FONT:SIZE
20, indicating the font size is 20 points.

DCH<x>_D<x>:LABe1:FONT:TYPE

This command sets or queries the font type of the label of the specified digital bit, such as Arial or Times New Roman.

Group Digital

Syntax DCH<x>_D<x>:LABe1:FONT:TYPE <QString>
DCH<x>_D<x>:LABe1:FONT:TYPE?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
 D<x> specifies the digital bits. The supported values are 0 to 15.
 <QString> is the font type of the label.

Examples DCH1_D1:LABel:FONT:TYPE "Monospace" sets the font to Monospace.
 DCH1_D1:LABel:FONT:TYPE? might return :DCH1_D1:LABel:FONT:TYPE
 "Frutiger LT Std 55 Roman", indicating the font is Frutiger LT Std 55
 Roman.

DCH<x>_D<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the label of the specified digital bit.

Group Digital

Syntax DCH<x>_D<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}
 DCH<x>_D<x>:LABel:FONT:UNDERline?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
 D<x> specifies the digital bits. The supported values are 0 to 15.
 <NR1> = 0 turns off underline font; any other value turns on italic font.
 OFF turns off underline font.
 ON turns on underline font.

Examples DCH1_D1:LABel:FONT:UNDERline ON turns on italic font.
 DCH1_D1:LABel:FONT:UNDERline? might return
 :DCH1_D1:LABel:FONT:UNDERline 0, indicating the font is not italic.

DCH<x>_D<x>:LABel:NAME

This command sets or queries the label of the specified digital bit.

Group Digital

Syntax DCH<x>_D<x>:LABeL:NAME <QString>
DCH<x>_D<x>:LABeL:NAME?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
D<x> specifies the digital bits. specifies the digital bits. The supported values are 0 to 15 or DALL (all digital bits).
<QString> is the label name.

Examples DCH1_D1:LABeL:NAME "Digital 1" sets the label name to Digital 1.
DCH1_D1:LABeL:NAME? might return :DCH1_D1:LABeL:NAME "Digital 1", indicating the label name is Digital 1.

*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a *TRG command. Define Device Trigger (*DDT) is a special alias that the *TRG command uses.

Group Miscellaneous

Syntax *DDT {<Block>|<QString>}
*DDT?

Related Commands [ALias](#)
[*TRG](#)

Arguments <Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters. The format of this argument is always returned as a query.
<QString> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.

Examples *DDT #217ACQUIRE:STATE RUN specifies that the acquisition system will be started each time a *TRG command is sent.

DESE

This command sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax DESE <NR1>
DESE?

Related Commands [*CLS](#)
[*ESE](#)
[*ESR?](#)
[EVENT?](#)
[EVMsg?](#)
[*SRE](#)
[*STB?](#)

Arguments <NR1> The binary bits of the DESER are set according to this value, which ranges from 1 through 255. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER maintains the previous power cycle value through the current power cycle.

NOTE. *Setting the DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the [*ESE](#) command to set the ESER.*

Examples DESE 209 sets the DESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.

DESE? might return DESE 186, showing that the DESER contains the binary value 10111010.

DIAG:LOOP:OPTion

This command sets or queries the type of looping desired.

Group	Self Test
Syntax	DIAG:LOOP:OPTion {FAIL ONCE ALWAYS ONFAIL NTIMES} DIAG:LOOP:OPTion?
Arguments	<p>Fail - run until a failure is found, then halt.</p> <p>Once - run through one loop.</p> <p>Always - run forever.</p> <p>Onfail - run until a failure is found, then loop on it.</p> <p>Ntimes - run n number of loops.</p>
Examples	<p>DIAG:LOOP:OPTION ALWAYS sets loop option to ALWAYS.</p> <p>DIAG:LOOP:OPTION? returns current looping option value.</p>

DIAG:LOOP:OPTion:NTIMes

This command sets or queries how many loops to run, if N-times is being used.

Group	Self Test
Syntax	DIAG:LOOP:OPTion:NTIMes <NR1> DIAG:LOOP:OPTion:NTIMes?
Arguments	<NR1> is how many loops to run.
Examples	<p>DIAG:LOOP:OPTION:NTIMES 2 sets diagnostics to loop 2 times.</p> <p>DIAG:LOOP:OPTION:NTIMES? might return DIAG:LOOP:OPTION:NTIMES 1 indicating diagnostics will run once.</p>

DIAG:LOOP:STOP (No Query Form)

Request that diagnostics stop looping.

Group	Self Test
Syntax	DIAG:LOOP:STOP
Examples	DIAG:LOOP:STOP stops diagnostics looping.

DIAG:MODE

This command sets or queries the diagnostics mode.

Group	Self Test
Syntax	DIAG:MODE {POST EXTENDED SERVICE} DIAG:MODE?
Arguments	POST specifies the power on self test diagnostics. EXTENDED specifies the extended diagnostics. SERVICE specifies the service diagnostics.
Examples	DIAG:MODE POST specifies the power on self test diagnostics.

DIAG:RESULT? (Query Only)

This query returns both the overall diagnostics test results and the results of each individual test area.

Group	Self Test
Syntax	DIAG:RESULT?
Returns	The diagnostics results.
Examples	DIAG:RESULT? might return DIAG:RESULT:FLAG "NOT RUN";LOG "PASS--IO, PASS--ANALOG, PASS--SYSTEM, PASS--ASIC, PASS--ACQ, PASS--SIGNAL, PASS--MEMORY" indicating the diagnostics results.

DIAG:RESUlt:FLAg? (Query Only)

This query returns the status of the diagnostic test area that has been selected.

Group Self Test

Syntax DIAG:RESUlt:FLAg?

Related Commands [DIAG:RESUlt:FLAg?](#)

Returns The status of the diagnostics (single area).

Examples DIAG:RESULT:FLAG? might return DIAG:RESULT:FLAG "NOT RUN" indicating the diagnostics have not been run.

DIAG:RESUlt:LOG? (Query Only)

This query returns the test Pass/Fail status of each diagnostic area. It does not return the overall status.

Group Self Test

Syntax DIAG:RESUlt:LOG?

Returns The status of the diagnostic area.

Examples DIAG:RESULT:LOG? might return :DIAG:RESULT:LOG "PASS--IO, PASS--ANALOG, PASS--SYSTEM, PASS--ASIC, PASS--ACQ, PASS--SIGNAL, PASS--MEMORY" indicating the diagnostics result stored in the log.

DIAG:SELEct (No Query Form)

This command selects or queries an available diagnostic area.

Group Self Test

Syntax	DIAG:SELEct {ALL IO ANALOG SYSTEM ASIC ACQ SIGNAL MEMORY} DIAG:SELEct?
Arguments	The argument is the desired diagnostic area.
Examples	DIAG:SELECT IO will select the IO group.

DIAG:STATE (No Query Form)

This command starts or aborts Self Test. Abort happens after group under test completes.

Group	Self Test
Syntax	DIAG:STATE {EXECute ABOrt}
Arguments	EXECUTE starts execution of the diagnostics. ABOrt disables diagnostics capabilities and returns the instrument to a normal operating state.
Examples	DIAG:STATE ABOrt turns off diagnostics capabilities and returns the instrument to a normal operating state. DIAG:STATE? might return DIAG:STATE ABOrt, indicating that diagnostics are disabled.

DISplay? (Query Only)

This query-only command returns the current Display settings.

Group	Display Control
Syntax	DISplay?
Examples	DISPLAY? might return DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW MONOGREEN; RECORDVIEW TEMPERATURE;USER:GRATICULE 165,50,15;CH1 180,50,100; CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0; REF2

```
0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1
160,50,100; MATH2 40,60,100;MATH3 120,60,100;MATH4
195,50,100; HISTOGRAM 320,50,100;CARET 150,50,100;MASK
0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR
DEFAULT;REFCOLOR DEFAULT; :DISPLAY:FILTER SINX;FORMAT
YT;GRATICULE IRE;INTENSITY :WAVEFORM:IMAGEVIEW
81.0000;RECORDVIEW 81.0000;:DISPLAY :INTENSITY:AUTOBRIGHT
0; :DISPLAY:PERSISTENCE OFF;STYLE DOTS;TRIGBAR OFF;TRIGT
1; CURSORTICK LONG;VARPERSIST 2.6000;SCREENTEXT:STATE
1; LABEL1:NAME "";XPOS 100;YPOS 5;:DISPLAY:SCREENTEXT
:LABEL2:NAME "THIS IS SCREEN TEXT";XPOS 100;YPOS 20;
:DISPLAY:SCREENTEXT:LABEL3:NAME "";XPOS 100;YPOS 35;
:DISPLAY:SCREENTEXT:LABEL4:NAME "";XPOS 100;YPOS 50;
:DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS 100;YPOS 343;
:DISPLAY:SCREENTEXT:LABEL6:NAME "";XPOS 100;YPOS 80;
:DISPLAY:SCREENTEXT:LABEL7:NAME "";XPOS 100;YPOS 95;
:DISPLAY:SCREENTEXT:LABEL8:NAME "";XPOS 100;YPOS 110;
:DISPLAY:WAVEFORM 1.
```

DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor

This command sets or queries the Inverted mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default Inverted colors and remain in effect until you reset the colors.

Group	Display
Syntax	DISplay:{CH<x> Math<x> REF<x>}:INVERTColor COLOR<y> DISplay:{CH<x> Math<x> REF<x>}:INVERTColor?
Arguments	<p>CH<x> specifies the input channel for which you want to change the waveform color, where <x> is the channel number.</p> <p>Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.</p> <p>REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.</p> <p>COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.</p>
Examples	DISplay:Ch1:INVERTColor COLOR23 sets the channel 2 Inverted waveform color to the value 23.

DISplay:REF3:INVERTColor? might return COLOR2, indicating that the Inverted waveform color setting for reference waveform 3 is 2.

DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor

This command sets or queries the normal mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default normal colors and remain in effect until you reset the colors.

Group	Display
Syntax	DISplay:{CH<x> Math<x> REF<x>}:NORMALColor COLOR<y> DISplay:{CH<x> Math<x> REF<x>}:NORMALColor?
Arguments	<p>CH<x> specifies the input channel for which you want to change the waveform color, where <x> is the channel number.</p> <p>Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.</p> <p>REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.</p> <p>COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.</p>
Examples	<p>DISplay:Ch1:NORMALColor COLOR13 sets the channel 2 normal waveform color to the value 13.</p> <p>DISplay:REF3:NORMALColor? might return COLOR12, indicating that the normal waveform color setting for reference waveform 3 is 12.</p>

DISplay:COLors

Sets or queries the color mode for the graticule and waveform display.

Group	Display Control
Syntax	DISplay:COLors {NORMAl INVERTed} DISplay:COLors?

- Arguments** `NORMAL` specifies normal color mode.
 `INVERTED` specifies inverted color mode.
- Examples** `DISPLAY:COLORS NORMAL` sets the display colors to normal.
 `DISPLAY:COLORS?` might return `DISPLAY:COLORS INVERTED` indicating the display colors are inverted.

DISplay:GLObal:B<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified bus. Setting this value true (On or $NR1 \neq 0$) turns on the source in the waveform view. Setting this value false (Off or $NR1 = 0$) turns off the source in the waveform view. This command only works if the specified bus is added already.

- Group** Display Control
- Syntax** `DISplay:GLObal:B<x>:STATE {<NR1>|OFF|ON}`
- Arguments** `<NR1> = 0` disables the display of the specified bus; any other value enables display of the bus.

 `ON` enables display of the specified bus.

 `OFF` disables display of the specified bus.
- Examples** `DISplay:GLObal:B1:STATE OFF` turns off the display of bus 1.

 `DISplay:GLObal:B1:STATE?` might return `DISPLAY:GLOBAL:B1:STATE 1` indicating the bus is displayed.

DISplay:GLObal:CH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified channel (both analog and digital). Setting this value true (On or $NR1 \neq 0$) turns on the source in the waveform view. Setting this value false (Off or $NR1 = 0$) turns off the source in the waveform view. This command only works if the specified channel is added already.

- Group** Display Control

Syntax	<code>DISPlay:GLObal:CH<x>:STATE {<NR1> OFF ON}</code>
Arguments	<p><NR1> = 0 disables the display of the specified channel; any other value enables display of the channel.</p> <p>ON enables display of the specified channel.</p> <p>OFF disables display of the specified channel.</p>
Examples	<p><code>DISPlay:GLObal:CH1:STATE OFF</code> turns off the display of channel 1.</p> <p><code>DISPlay:GLObal:CH1:STATE?</code> might return <code>:DISPLAY:GLOBAL:CH1:STATE 0</code> indicating that channel 1 is not displayed.</p>

DISPlay:GLObal:DCH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified channel (digital). Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified channel is added already.

Group	Display Control
Syntax	<code>DISPlay:GLObal:DCH<x>:STATE {<NR1> OFF ON}</code>
Arguments	<p>DCH<x> = specifies the digital channel. The supported value is 1.</p> <p><NR1> = 0 disables the display of the specified digital channel; any other value enables display of the channel.</p> <p>ON enables display of the specified digital channel.</p> <p>OFF disables display of the specified digital channel.</p>
Examples	<p><code>DISPlay:GLObal:DCH1:STATE OFF</code> turns off the display of digital channel 1.</p> <p><code>DISPlay:GLObal:DCH1:STATE?</code> might return <code>:DISPLAY:GLOBAL:DCH1:STATE 0</code> indicating that digital channel 1 is not displayed.</p>

DISplay:GLObal:MATH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified math. Setting this value true (On or NR1 \neq 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified math waveform is added already.

Group Display Control

Syntax DISplay:GLObal:MATH<x>:STATE {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables the display of the specified math; any other value enables display of the math.

ON enables display of the specified math.

OFF disables display of the specified math.

Examples DISplay:GLObal:MATH1:STATE 1 displays math 1.

DISplay:GLObal:MATH1:STATE? might return
:DISPLAY:GLOBAL:MATH1:STATE 1 indicating that math1 is displayed.

DISplay:GLObal:REF<x>:STATE

this command sets or queries the global state (display mode On or Off) of the specified reference waveform. Setting this value true (On or NR1 \neq 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified reference waveform is added already.

Group Display Control

Syntax DISplay:GLObal:REF<x>:STATE {<NR1>|OFF|ON}

Arguments REF<x> is the Reference waveform number.

<NR1> = 0 disables the display of the specified reference; any other value enables display of the reference.

ON enables display of the specified reference.

OFF disables display of the specified reference.

Examples `DISPlay:GLObal:REF1:STATE 1` displays reference 1.
 `DISPlay:GLObal:REF1:STATE?` might return
 `:DISPLAY:GLOBAL:REF1:STATE 1` indicating that reference 1 is displayed.

DISPlay:INTENSITY? (Query Only)

This query-only command returns the waveform saturation level and screen saver settings.

Group Display Control

Syntax `DISPlay:INTENSITY?`

Related Commands

Arguments None

Examples `DISPLAY:INTENSITY?` might return `:DISPLAY:INTENSITY:BACKLIGHT 1`, indicating the intensity is set to 1.

DISPlay:INTENSITY:BACKLight

This command sets or queries the display backlight intensity setting.

Group Display Control

Syntax `DISPlay:INTENSITY:BACKLight {LOW|MEDIum|HIGH}`
`DISPlay:INTENSITY:BACKLight?`

Arguments `LOW` selects a low brightness level.
 `MEDIum` selects a moderate brightness level.
 `HIGH` selects a full brightness level.

Examples `DISPLAY:INTENSITY:BACKLIGHT LOW` sets the display backlight to low brightness level.

DISPLAY:INTENSITY:BACKLIGHT? might return
DISPLAY:INTENSITY:BACKLIGHT HIGH, indicating that the display backlight
is set to full brightness level.

DISplay:INTENSITy:BACKLight:AUTODim:ENABle

Sets or queries the state of the display auto-dim feature. The default is enabled. Once the backlight has dimmed, any button push, knob turn or mouse movement returns the backlight value to the value set by :DISplay:INTENSITy:BACKLight.

Group Display Control

Syntax DISplay:INTENSITy:BACKLight:AUTODim:ENABle {ON|OFF}
DISplay:INTENSITy:BACKLight:AUTODim:ENABle?

Related Commands [DISplay:INTENSITy:BACKLight](#)

Arguments ON enables the display auto-dim feature.
OFF disables the display auto-dim feature.

Examples DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON enables
auto-dimming of the backlight.
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE? might return
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON, indicating that
auto-dimming of the backlight is enabled.

DISplay:INTENSITy:BACKLight:AUTODim:TIME

Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display. The time can range from a minimum of 10 minutes to a maximum of 1440 minutes (24 hours). The default is 10 minutes.

Group Display Control

Syntax DISplay:INTENSITy:BACKLight:AUTODim:TIME <NR1>
DISplay:INTENSITy:BACKLight:AUTODim:TIME?

Arguments <NR1> is the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.

- Examples** `DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 30` sets the backlight to autodim in 30 minutes.
- `DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME?` might return `DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 60`, indicating the backlight will auto-dim in 60 minutes.

DISplay:MATHFFTView<n>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

- Group** Cursor
- Syntax** `DISplay:MATHFFTView<n>:CURSor:ROLOCATION {GRATICULE|BADGE}`
`DISplay:MATHFFTView<n>:CURSor:ROLOCATION?`
- Arguments** `MATHFFTView<x>` is the Math FFT plot number.
- `GRATICULE` sets the Math FFT plot cursor readouts to display as part of the cursors in the plot view.
- `BADGE` removes the Math FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.
- Examples** `DISplay:MATHFFTView4:CURSor:ROLOCATION GRATICULE` sets the cursor readout information for Math FFT plot 4 to display in the plot graticule.
- `DISplay:MATHFFTView2:CURSor:ROLOCATION?` might return `DISplay:MATHFFTView2:CURSor:ROLOCATION BADGE`, indicating that cursor readout information for Math FFT plot 2 is set to display as a badge in the Results Bar.

DISplay:MATHFFTView<x>:AUTOScale

This command sets or returns the enabled state of autoscale for Math/FFT waveforms.

- Group** Display Control
- Syntax** `DISplay:MATHFFTView<x>:AUTOScale {OFF|ON|<NR1>}`
`DISplay:MATHFFTView<x>:AUTOScale?`

Arguments MATHFFTView<x> is the Math-FFT waveform number.
OFF disables the autoscale feature.
ON enables the autoscale feature.
<NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples DISPlay:MATHFFTView1:AUTOScale OFF disables the autoscale feature.
DISPlay:MATHFFTView3:AUTOScale? might return
DISPLAY:MATHFFTVIEW3:AUTOSCALE 1, indicating that autoscale
is on for the Math3 FFT waveform.

DISPlay:MATHFFTView<x>:CURSor:ASOURce? (Query Only)

This command queries the Math-FFT waveform view source for cursor A.

Group Cursor

Syntax DISPlay:MATHFFTView<x>:CURSor:ASOURce?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISPlay:MATHFFTView1:CURSor:ASOURce? might return
DISPLAY:MATHFFTVIEW1:CURSOR:ASOURCE MATH1, indicating the source
is MATH 1.

DISPlay:MATHFFTView<x>:CURSor:BSOURce? (Query Only)

This command queries the Math-FFT waveform view source for cursor B.

Group Cursor

Syntax DISPlay:MATHFFTView<x>:CURSor:BSOURce?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView5:CURSor:BSOURCE?` might return
`DISPLAY:MATHFFTVIEW5:CURSOR:BSOURCE MATH5`, indicating the source
is Math waveform MATH 5.

DISPlay:MATHFFTView<x>:CURSor:DDT? (Query Only)

This command queries the delta Y over delta X ($\Delta Y/\Delta X$) cursor readout value of the specified cursor in the specified Math-FFT view.

NOTE. *If the current cursor is set to horizontal mode, this command will time out.*

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:DDT?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView<x>:CURSor:DDT?` might return
`DISPLAY:MATHFFTVIEW1:CURSOR:DDT 14.0056E-9`, indicating
the $\Delta Y/\Delta X$ cursor readout value is 14.0056E-9.

DISPlay:MATHFFTView<x>:CURSor:FUNCTION

This command sets or queries the cursor type for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:FUNCTION`
`{WAVEform|VBArS|HBArS|SCREEN}`
`DISPlay:MATHFFTView<x>:CURSor:FUNCTION?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`WAVEFORM` specifies to display the paired vertical cursors in YT display format for measuring waveform amplitude and time. Measurements are taken at where the cursor intersects the waveform, and tracks waveform changes.

`VBArS` specifies vertical bar cursors, which measure in horizontal units.

`HBArS` specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which display the horizontal and vertical positions of the cursors, not waveform levels. Use these cursors to measure anywhere in the waveform display area.

Examples `DISPlay:MATHFFTView5:CURSor:FUNctIon HBARS` sets Math-FFT waveform 5 to use horizontal bar cursors.

`DISPlay:MATHFFTView1:CURSor:FUNctIon?` might return
`DISPlay:MATHFFTView1:CURSor:FUNctIon? SCREEN`, indicating that Math-FFT waveform 2 is set to use both horizontal and vertical cursors.

DISPlay:MATHFFTView<x>:CURSor:HBArS:APOSition

This command sets or returns the position of horizontal cursor A for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:HBArS:APOSition <NR3>`
`DISPlay:MATHFFTView<x>:CURSor:HBArS:APOSition?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.
`<NR3>` is the cursor position of the specified cursor in the specified view.

Examples `DISPlay:MATHFFTView1:CURSor:HBArS:APOSition 50.0e-3` sets the position to 50 mV.

`DISPlay:MATHFFTView4:CURSor:HBArS:APOSition?` might return
`DISPLAY:MATHFFTVIEW4:CURSOR:HBARS:APOSITION -274.2247E-3`,
indicating the A cursor position is $-274.2247E-3$.

DISPlay:MATHFFTView<x>:CURSor:HBArS:AUNIts? (Query Only)

This command queries the vertical units of horizontal cursor A for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:HBArS:AUNIts?`

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISPlay:MATHFFTView1:CURSor:HBARs:AUNIts? might return
 DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:AUNITS "dBm", indicating the
 vertical units are dBm.

DISPlay:MATHFFTView<x>:CURSor:HBARs:BPOStion

This command sets or returns the position of horizontal cursor B for the specified Math-FFT view.

Group Cursor

Syntax DISPlay:MATHFFTView<x>:CURSor:HBARs:BPOStion <NR3>
 DISPlay:MATHFFTView<x>:CURSor:HBARs:BPOStion?

Arguments MATHFFTView<x> is the Math-FFT waveform number.
 <NR3> is the vertical cursor B position for the specified Math-FFT view.

Examples DISPlay:MATHFFTView1:CURSor:HBARs:BPOStion 50e-3 sets the
 position to 0.005 vertical units.
 DISPlay:MATHFFTView1:CURSor:HBARs:BPOStion? might return
 DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:BPOSITION -40.4061, indicating
 the position is -40.4061 vertical units.

DISPlay:MATHFFTView<x>:CURSor:HBARs:BUNIts? (Query Only)

This command queries the vertical units of horizontal cursor B for the specified Math-FFT view.

Group Cursor

Syntax DISPlay:MATHFFTView<x>:CURSor:HBARs:BUNIts?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView3:CURSor:HBARs:BUNIts?` might return
`DISPLAY:MATHFFTVIEW3:CURSOR:HBARS:BUNITS "dBm"`, indicating the
cursor units are dBm.

DISPlay:MATHFFTView<x>:CURSor:HBARs:DELTA? (Query Only)

This command queries the horizontal cursor's delta value of the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:HBARs:DELTA?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView1:CURSor:HBARs:DELTA?` might return
`DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:DELTA 30.61448`, indicating the
cursor readout is 30.61488 units.

DISPlay:MATHFFTView<x>:CURSor:MODE

This command sets or queries the cursor tracking mode of the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:MODE {INDEPENDENT|TRACK}`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`INDEPENDENT` allows independent adjustment of the two cursors.

`TRACK` ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

Examples `DISPlay:MATHFFTView1:CURSor:MODE INDEPENDENT` allows independent
adjustment of the two cursors.

DISPlay:MATHFFTView1:CURSOR:MODE? might return
:DISPLAY:MATHFFTVIEW1:CURSOR:MODE INDEPENDENT, indicating the mode
is set to independent.

DISPlay:MATHFFTView<x>:CURSOR:ONEOVERDELTAVALUE? (Query Only)

This command queries the one over delta cursor readout value of the specified Math-FFT view.

Group	Cursor
Syntax	DISPlay:MATHFFTView<x>:CURSOR:ONEOVERDELTAVALUE?
Arguments	MATHFFTView<x> is the Math-FFT waveform number.
Examples	DISPlay:MATHFFTView1:CURSOR:ONEOVERDELTAVALUE? might return DISPLAY:MATHFFTVIEW1:CURSOR:ONEOVERDELTAVALUE 2.6005E-9, indicating the one over delta time value is 2.6005 nanoseconds.

DISPlay:MATHFFTView<x>:CURSOR:SCREEN:AXPOSITION

This command sets or returns the vertical cursor A x-axis waveform measurement position of the specified Math-FFT view.

Group	Cursor
Syntax	DISPlay:MATHFFTView<x>:CURSOR:SCREEN:AXPOSITION <NR3> DISPlay:MATHFFTView<x>:CURSOR:SCREEN:AXPOSITION?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. <NR3> is the cursor position in MHz.
Examples	DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:AXPOSITION 700.000E+6 sets the cursor position to 700 MHz. DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION? might return DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION 125.000E+6, indicating the x-axis value of cursor A is 125 MHz.

DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOsition

This command sets or returns the vertical cursor A y-axis amplitude measurement value of the specified Math-FFT view.

Group	Cursor
Syntax	DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOsition <NR3> DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOsition?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. <NR3> is the cursor A position of the specified cursor in the specified view.
Examples	DISplay:MATHFFTView2:CURSor:SCREEN:AYPOsition 1.0e0 sets the cursor position to 1.0 dBm. DISplay:MATHFFTView5:CURSor:SCREEN:AYPOsition? might return DISPLAY:MATHFFTVIEW5:CURSOR:SCREEN:AYPOSITION 1.1741714106633, indicating the cursor position is 1.174 vertical units (such as dBm).

DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOsition

This command sets or returns the vertical cursor Bx-axis waveform time measurement position of the specified Math-FFT view.

Group	Cursor
Syntax	DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOsition <NR3> DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOsition?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. <NR3> is the horizontal cursor B position of the specified cursor in the specified view.
Examples	DISplay:MATHFFTView1:CURSor:SCREEN:BXPOsition 3.5e9 sets the cursor position to 3.5 GHz.

DISPlay:MATHFFTView1:CURSOR:SCREEN:BXPOSITION? might return
 DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:BXPOSITION 2.50E+9,
 indicating the cursor position is 2.5 GHz.

DISPlay:MATHFFTView<x>:CURSOR:SCREEN:BYPOSITION

This command sets or returns the vertical cursor B y-axis amplitude measurement value of the specified Math-FFT view.

Group	Cursor
Syntax	DISPlay:MATHFFTView<x>:CURSOR:SCREEN:BYPOSITION <NR3> DISPlay:MATHFFTView<x>:CURSOR:SCREEN:BYPOSITION?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. <NR3> is the vertical cursor B position of the specified cursor in the specified view.
Examples	DISPlay:MATHFFTView1:CURSOR:SCREEN:BYPOSITION -50.0e0 sets the cursor position to -50.0. DISPlay:MATHFFTView4:CURSOR:SCREEN:BYPOSITION? might return DISPLAY:MATHFFTVIEW4:CURSOR:SCREEN:BYPOSITION -59.0460929065173, indicating the cursor position is at -59.0 dBm.

DISPlay:MATHFFTView<x>:CURSOR:STATE

This command sets or queries the visible state of cursors for the specified Math-FFT view.

Group	Cursor
Syntax	DISPlay:MATHFFTView<x>:CURSOR:STATE {ON OFF <NR1>} DISPlay:MATHFFTView<x>:CURSOR:STATE?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. 1 or ON enables the cursors. 0 or OFF disables the cursors. <NR1> = 0 turns off cursors; any other value displays cursors.

Examples `DISPlay:MATHFFTView2:CURSor:STATE OFF` turns off the cursors for Math-FFT waveform 2.

`DISPlay:MATHFFTView1:CURSor:STATE?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:STATE 1` indicating the cursors are enabled.

DISPlay:MATHFFTView<x>:CURSor:VBArS:APOsition

This command sets or queries the horizontal cursor A position for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:VBArS:APOsition <NR3>`
`DISPlay:MATHFFTView<x>:CURSor:VBArS:APOsition?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` sets the vertical cursor A position in the specified view.

Examples `DISPlay:MATHFFTView1:CURSor:VBArS:APOsition 500.0e6` sets the cursor A position to 500 MHz.

`DISPlay:MATHFFTView1:CURSor:VBArS:APOsition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:APOSITION 585.533869115958E+6`, indicating cursor position is 585.5 MHz.

DISPlay:MATHFFTView<x>:CURSor:VBArS:AUNItS? (Query Only)

This command queries the vertical cursor A measurement units for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:VBArS:AUNItS?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView1:CUSor:VBArS:AUNITs?` might return `:DISPlay:MATHFFTVIEW1:CUSor:VBArS:UNITs "dBm"`, indicating the cursor units are dBm.

`DISPlay:MATHFFTView<x>:CURSor:VBArS:BPOsition`

This command sets or queries the vertical cursor B position for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:VBArS:BPOsition <NR3>`
`DISPlay:MATHFFTView<x>:CURSor:VBArS:BPOsition?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.
`<NR3>` sets the vertical cursor B position in the specified view.

Examples `DISPlay:MATHFFTView1:CUSor:VBArS:BPOsition 2.0e9` sets the cursor B position to 2.0 GHz.
`DISPlay:MATHFFTView1:CUSor:VBArS:BPOsition?` might return `DISPlay:MATHFFTVIEW1:CUSor:VBArS:BPOsITION 2.50E+9`, indicating the cursor position is at 2.5 GHz.

`DISPlay:MATHFFTView<x>:CURSor:VBArS:BUNITs? (Query Only)`

This command queries the vertical cursor B measurement units for the specified Math-FFT view.

Group Cursor

Syntax `DISPlay:MATHFFTView<x>:CURSor:VBArS:BUNITs?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISPlay:MATHFFTView1:CUSor:VBArS:BUNITs?` might return `:DISPlay:MATHFFTVIEW1:CUSor:VBArS:BUNITs "Hz"` indicating the cursor units are Hz.

DISplay:MATHFFTView<x>:CURSor:VBArS:DELTA? (Query Only)

This command queries the vertical cursor's delta T readout value for the specified Math-FFT view.

Group	Cursor
Syntax	DISplay:MATHFFTView<x>:CURSor:VBArS:DELTA?
Arguments	MATHFFTView<x> is the Math-FFT waveform number.
Examples	DISplay:MATHFFTView1:CURSor:VBArS:DELTA? might return DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:DELTA 1.9144661308840E+9, indicating the delta T cursor value is 1.9 GHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition

This command sets or queries the waveform cursor A position for the specified Math-FFT view.

Group	Cursor
Syntax	DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition <NR3> DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition?
Arguments	MATHFFTView<x> is the Math-FFT waveform number. <NR3> is the waveform cursor A position in the specified plot view.
Examples	DISplay:MATHFFTView1:CURSor:WAVEform:APOSition 600.0e6 sets the position to 600 MHz. DISplay:MATHFFTView1:CURSor:WAVEform:APOSition? might return DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:APOSITION 25.000E+6, indicating the cursor position is 25 MHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B position for the specified Math-FFT view.

Group	Cursor
Syntax	<pre>DISPlay:MATHFFTView<x>:CURSOR:WAVEform:BPOSITION <NR3> DISPlay:MATHFFTView<x>:CURSOR:WAVEform:BPOSITION?</pre>
Arguments	<p>MATHFFTView<x> is the Math-FFT waveform number.</p> <p><NR3> is the waveform cursor B position in the specified plot view.</p>
Examples	<pre>DISPlay:MATHFFTView1:CURSOR:WAVEform:BPOSITION 2.0E+9</pre> sets the cursor position to 2.0 GHz. <p><pre>DISPlay:MATHFFTView1:CURSOR:WAVEform:BPOSITION?</pre> might return <pre>DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:BPOSITION 250.0E+6</pre>, indicating the waveform cursor B position is 250 MHz.</p>

DISPlay:MATHFFTView<x>:GRIDlines

This command sets or queries the grid lines setting for the specified Math-FFT view.

Group	Display Control
Syntax	<pre>DISPlay:MATHFFTView<x>:GRIDlines {HORIZONTAL VERTICAL BOTH} DISPlay:MATHFFTView<x>:GRIDlines?</pre>
Arguments	<p>MATHFFTView<x> is the Math-FFT waveform number.</p> <p>HORIZONTAL specifies horizontal grid lines.</p> <p>VERTICAL specifies vertical grid lines.</p> <p>BOTH specifies both vertical and horizontal grid lines.</p>
Examples	<pre>DISPlay:MATHFFTView1:GRIDlines VERTICAL</pre> specifies vertical grid lines. <p><pre>DISPlay:MATHFFTView1:GRIDlines?</pre> might return <pre>:DISPLAY:MATHFFTVIEW1:GRIDLINES BOTH</pre> indicating both horizontal and vertical grid lines are displayed.</p>

DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE

This command sets or queries the display state of the specified math waveform for the specified Math-FFT view.

Group Display Control

Syntax DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE {OFF|ON|<NR1>}

Arguments MATHFFTView<x> is the Math-FFT waveform number.
OFF disables displaying the specified Math-FFT view.
ON enables displaying the specified Math-FFT view.
<NR1> = 0 disables the specified Math-FFT view; any other value enables the specified Math-FFT view.

Examples DISplay:MATHFFTView1:MATH:MATH1:STATE ON enables the specified Math-FFT view.
DISplay:MATHFFTView4:MATH:MATH4:STATE? might return
:DISPLAY:MATHFFTVIEW1:MATH:MATH1:STATE 1, indicating the waveform Math-FFT 1 is displayed.

DISplay:MATHFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale (Linear or Log) for the specified Math-FFT view.

Group Display Control

Syntax DISplay:MATHFFTView<x>:XAXIS:SCALE {LINEAR|LOG}
DISplay:MATHFFTView<x>:XAXIS:SCALE?

Arguments MATHFFTView<x> is the Math-FFT waveform number.
LINEAR specifies a linear scale.
LOG specifies a logarithmic scale.

Examples `DISPlay:MATHFFTView1:XAXIS:SCALE LOG` specifies a logarithmic scale.
`DISPlay:MATHFFTView1:XAXIS:SCALE?` might return
`DISPLAY:MATHFFTVIEW1:XAXIS:SCALE LINEAR`, indicating that
the scale setting is linear.

DISPlay:MATHFFTView<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting (Linear or dBm) for the specified Math-FFT view.

Group Display Control

Syntax `DISPlay:MATHFFTView<x>:YAXIS:SCALE {LINEAR|DBM}`
`DISPlay:MATHFFTView<x>:YAXIS:SCALE?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`LINEAR` specifies a linear scale.

`DBM` specifies a dBm scale.

Examples `DISPlay:MATHFFTView1:YAXIS:SCALE LINEAR` specifies a linear scale.
`DISPlay:MATHFFTView1:YAXIS:SCALE?` might return
`DISPLAY:MATHFFTVIEW1:YAXIS:SCALE DBM`, indicating that the
scale is dBm.

DISPlay:MATHFFTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the zoom area for the specified Math-FFT view.

Group Zoom

Syntax `DISPlay:MATHFFTView<x>:ZOOM:XAXIS:FROM <NR3>`
`DISPlay:MATHFFTView<x>:ZOOM:XAXIS:FROM?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the value of the left edge of the zoom x axis in the specified plot view.

Examples `DISPlay:MATHFFTView1:ZOOM:XAXIS:FROM 800.e6` sets the x axis value of the left edge of the plot view to 800 MHz.

`DISPlay:MATHFFTView1:ZOOM:XAXIS:FROM?` might return
`DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:FROM 781.0E+6`, indicating the x axis value of the left edge of the plot view is at 781 MHz.

DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge value of the zoom area for the specified Math-FFT view.

Group Zoom

Syntax `DISPlay:MATHFFTView<x>:ZOOM:XAXIS:TO <NR3>`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.
`<NR3>` is the value of the right edge of the zoom x axis in the specified plot view.

Examples `DISPlay:MATHFFTView1:ZOOM:XAXIS:TO 2.3E+9` sets the value of the right edge of the x axis to 2.3 GHz.

`DISPlay:MATHFFTView1:ZOOM:XAXIS:TO?` might return
`:DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:TO 2.343750E+9`,
indicating the value of the right edge of the zoom x axis is 2.34 GHz.

DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom edge value of the zoom y-axis area for the specified Math-FFT view.

Group Zoom

Syntax `DISPlay:MATHFFTView<x>:ZOOM:YAXIS:FROM <NR3>`
`DISPlay:MATHFFTView<x>:ZOOM:YAXIS:FROM?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.
`<NR3>` is the bottom value of the zoom y axis in the specified plot view.

Examples `DISPlay:MATHFFTView1:ZOOM:YAXIS:FROM -75.0e0` sets the bottom value of the y axis to -75 dBm.

`DISPlay:MATHFFTView1:ZOOM:YAXIS:FROM?` might return
`DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:FROM -78.8258285893367`
 indicating the bottom value of the y axis is -78.8 dBm.

DISPlay:MATHFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.

Group Zoom

Syntax `DISPlay:MATHFFTView<x>:ZOOM:YAXIS:TO <NR3>`
`DISPlay:MATHFFTView<x>:ZOOM:YAXIS:TO?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.
`<NR3>` is the top value of the zoom y axis in the specified plot view.

Examples `DISPlay:MATHFFTView1:ZOOM:YAXIS:TO 21.0E0` sets the top value of the zoom y axis in the specified plot view to 21 dBm.

`DISPlay:MATHFFTView1:ZOOM:YAXIS:TO?` might return
`DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:TO 21.1741714106633,`
 indicating the top value of the zoom y axis in the specified plot view is 21.17 dBm.

DISPlay:PERSistence

This command sets or queries the display persistence for analog waveforms. Persistence is valid for wave views only.

Group Display Control

Syntax `DISPlay:PERSistence`
`{OFF|AUTO|INFPersist|INFinite|VARpersist|CLEAR}`
`DISPlay:PERSistence?`

Related Commands [DISPlay:VARpersist](#)

Arguments	<p>OFF disables the persistence aspect of the display.</p> <p>AUTO automatically set the persistence.</p> <p>INFPersist sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.</p> <p>INFINite sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.</p> <p>VARPersist sets a display mode where set pixels are gradually dimmed.</p> <p>CLEAR resets the persist time count down and clears the display of acquired points.</p>
Examples	<p>DISPLAY:PERSISTENCE VARPERSIST sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARpersist command.</p> <p>DISPLAY:PERSISTENCE? might return :DISPLAY:PERSISTENCE OFF, indicating that the persistence aspect of the display is disabled.</p>

DISplay:PERStence:RESEt (No Query Form)

This command controls the clearing of persistence data that has been built up over time. Persistence is valid for wave views only.

Group	Display Control
Syntax	DISplay:PERStence:RESEt
Related Commands	DISplay:PERStence DISplay:VARpersist
Examples	DISPLAY:PERSISTENCE:RESEt clears the display of persistence data.

DISplay:PLOTView<n>:CURSor:ROLOCAtion

This command sets or queries the location to display the specified plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group	Cursor
--------------	--------

Syntax	<pre>DISPlay:PLOTView<n>:CURSOR:ROLOCATION {GRATICULE BADGE} DISPlay:PLOTView<n>:CURSOR:ROLOCATION?</pre>
Arguments	<p>PLOTView<x> is the waveform plot number.</p> <p>GRATICULE sets the plot cursor readouts to display as part of the cursors in the plot view.</p> <p>BADGE removes the plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.</p>
Examples	<p>DISPlay:PLOTView1:CURSOR:ROLOCATION GRATICULE sets the cursor readout information for plot 1 to display in the plot graticule.</p> <p>DISPlay:PLOTView2:CURSOR:ROLOCATION? might return DISPlay:PLOTView2:CURSOR:ROLOCATION BADGE, indicating that cursor readout information for plot 2 is set to display as a badge in the Results Bar.</p>

DISPlay:PLOTView<x>:AUTOScale

This command sets or queries the enabled state of autoscale for the specified plot.

Group	Display Control
Syntax	<pre>DISPlay:PLOTView<x>:AUTOScale {OFF ON <NR1>} DISPlay:PLOTView<x>:AUTOScale?</pre>
Arguments	<p>PLOTView<x> is the Plot waveform number.</p> <p>OFF disables the autoscale feature.</p> <p>ON enables the autoscale feature.</p> <p><NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.</p>
Examples	<p>DISPlay:PLOTView1:AUTOScale OFF turns off plot autoscale.</p> <p>DISPlay:PLOTView3:AUTOScale? might return :DISPLAY:PLOTVIEW3:AUTOSCALE 0, indicating the plot autoscale is off for plot 3.</p>

DISplay:PLOTView<x>:CURSor:ASOURce? (Query Only)

This command queries the cursor source for plot cursor A.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:ASOURce?
Arguments	PLOTView<x> is the Plot waveform number.
Examples	DISplay:PLOTView1:CURSor:ASOURce? might return DISPLAY:PLOTVIEW1:CURSOR:ASOURCE PLOT1 indicating the source of the cursor is plot 1.

DISplay:PLOTView<x>:CURSor:BSOURce? (Query Only)

This command queries the cursor source for plot cursor B.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:BSOURce?
Arguments	PLOTView<x> is the Plot waveform number.
Examples	DISplay:PLOTView1:CURSor:BSOURce? might return :DISPLAY:PLOTVIEW1:CURSOR:BSOURCE PLOT1 indicating the B cursor source is plot 1.

DISplay:PLOTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value for the specified Plot view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:DDT?

Arguments `PlotView<x>` is the Plot waveform number.

Examples `DISplay:PLOTView1:CURSor:DDT?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:DDT 131.0470E+6` indicating the cursor
readout value is 131.0470E+6.

DISplay:PLOTView<x>:CURSor:FUNCtion

This command sets or queries the cursor mode for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:FUNCtion`
`{WAVEFORM|VBARS|HBARS|SCREEN}`
`DISplay:PLOTView<x>:CURSor:FUNCtion?`

Arguments `PLOTView<x>` is the Plot waveform number.

`WAVEFORM` specifies to display the paired cursors in YT display format for measuring waveform amplitude and time.

`VBARS` specifies vertical bar cursors, which measure in horizontal units.

`HBARS` specifies horizontal bar cursors, which measure in vertical units.

`SCREEN` specifies to display both horizontal and vertical bar cursors, which measure the selected waveform in horizontal and vertical units. Use these cursors to measure anywhere in the waveform display area.

Examples `DISplay:PLOTView1:CURSor:FUNCtion VBARS` displays vertical bar cursors.

`DISplay:PLOTView1:CURSor:FUNCtion?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:FUNCTION SCREEN` indicating that
both horizontal and vertical bar cursors are displayed.

DISplay:PLOTView<x>:CURSor:HBARS:APOSition

This command sets or queries the horizontal cursor A position for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:HBArS:APOsition <NR3>`
`DISplay:PLOTView<x>:CURSor:HBArS:APOsition?`

Arguments `PLOTView<x>` is the Plot waveform number.
`<NR3>` is the cursor position.

Examples `DISplay:PLOTView<x>:CURSor:HBArS:APOsition 2.0` sets the cursor to 2 Volts.
`DISplay:PLOTView<x>:CURSor:HBArS:APOsition?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:HBARS:APOSITION 2.2186` indicating the HBArS cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArS:AUNIts? (Query Only)

This command queries the horizontal cursor A vertical units for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:HBArS:AUNIts?`

Returns `PLOTView<x>` is the Plot waveform number.

Examples `DISplay:PLOTView1:CURSor:HBArS:AUNIts?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:HBARS:AUNITS "V"`, indicating the A units are volts.

DISplay:PLOTView<x>:CURSor:HBArS:BPOsition

This command sets or queries the horizontal cursor B position for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:HBArS:BPOsition <NR3>`
`DISplay:PLOTView<x>:CURSor:HBArS:BPOsition?`

Arguments	PLOTview<x> is the Plot waveform number. <NR3> is the HBArS vertical position.
Examples	DISplay:PLOTview<x>:CURSor:HBArS:BPOsition 2.0 sets the cursor to 2 Volts. DISplay:PLOTview<x>:CURSor:HBArS:BPOsition? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BPOSITION 2.2186 indicating the HBArS cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArS:BUNIts? (Query Only)

This command queries the cursor B vertical units for the specified Plot view.

Group	Cursor
Syntax	DISplay:PLOTview<x>:CURSor:HBArS:BUNIts?
Arguments	PLOTview<x> is the Plot waveform number.
Examples	DISplay:PLOTview1:CURSor:HBArS:BUNIts? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BUNITS "V" indicating the units are Volts.

DISplay:PLOTView<x>:CURSor:HBArS:DELTA? (Query Only)

This command queries the delta V cursor readout value for the specified Plot view.

Group	Cursor
Syntax	DISplay:PLOTview<x>:CURSor:HBArS:DELTA?
Arguments	PLOTview<x> is the Plot waveform number.
Returns	Returns an <NR3> that is the delta V cursor value.

Examples `DISPlay:PLOTView1:CURSor:HBARS:DELTA?` might return
 `:DISPLAY:PLOTVIEW1:CURSOR:HBARS:DELTA 1.2543`, indicating the delta
 value is 1.2543 V.

DISPlay:PLOTView<x>:CURSor:MODE

This command sets or queries the cursor tracking mode for the specified Plot view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSor:MODE {INDEPENDENT|TRACK}`
 `DISPlay:PLOTView<x>:CURSor:MODE?`

Arguments `PLOTView<x>` is the Plot waveform number.

`INDEPENDENT` allows independent adjustment of the two cursors.

`TRACK` ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A.

Examples `DISPlay:PLOTView1:CURSor:MODE TRACK` sets the cursor to track together.

 `DISPlay:PLOTView1:CURSor:MODE?` might return
 `:DISPLAY:PLOTVIEW1:CURSOR:MODE INDEPENDENT`, indicating independent
 adjustment of the cursors is allowed.

DISPlay:PLOTView<x>:CURSor:ONEOVERDELTAVALUE? (Query Only)

This command sets or queries the one over delta T cursor readout value for the specified Plot view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSor:ONEOVERDELTAVALUE?`

Arguments `PLOTView<x>` is the Plot waveform number.

Returns Returns an <NR3> that is the one over delta T cursor readout value (it may not be delta T, depending on the plot units).

Examples `DISPlay:PLOTView<x>:CURSOR:ONEOVERDELTAVALUE?` might return `:DISPLAY:PLOTVIEW1:CURSOR:ONEOVERDELTAVALUE 88.2295E+6` indicating the 1 over delta time value is 88.2295 MHz.

`DISPlay:PLOTView<x>:CURSOR:SCREEN:AXPOSITION`

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSOR:SCREEN:AXPOSITION <NR3>`
`DISPlay:PLOTView<x>:CURSOR:SCREEN:AXPOSITION?`

Arguments `PLOTView<x>` is the Plot waveform number.
`<NR3>` is the horizontal cursor A position.

Examples `DISPlay:PLOTView1:CURSOR:SCREEN:AXPOSITION -5.0` sets the A X cursor position to -5.0 ns.
`DISPlay:PLOTView1:CURSOR:SCREEN:AXPOSITION?` might return `:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AXPOSITION -5.5460E-9` indicating the cursor A X position is -5.546 ns.

`DISPlay:PLOTView<x>:CURSOR:SCREEN:AYPOSITION`

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSOR:SCREEN:AYPOSITION <NR3>`
`DISPlay:PLOTView<x>:CURSOR:SCREEN:AYPOSITION?`

Arguments `PLOTView<x>` is the Plot waveform number.
`<NR3>` is the vertical cursor A position.

Examples `DISPlay:PLOTView1:CURSOR:SCREEN:AYPOSITION 2.0` set the A cursor Y position to 2.0 Volts.

DISplay:PLOTView1:CURSor:SCREEN:AYPosition? might return
:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AYPOSITION 1.9035 indicating
the A cursor Y position is 1.9035 V.

DISplay:PLOTView<x>:CURSor:SCREEN:BXPosition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:SCREEN:BXPosition <NR3> DISplay:PLOTView<x>:CURSor:SCREEN:BXPosition?
Arguments	PLOTView<x> is the Plot waveform number. <NR3> is the horizontal cursor B position.
Examples	DISplay:PLOTView1:CURSor:SCREEN:BXPosition 59.0E-9 sets the B cursor X position to 59.0 ns. DISplay:PLOTView1:CURSor:SCREEN:BXPosition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BXPOSITION 59.6802E-9 indicating the B cursor X position is 59.6802 ns.

DISplay:PLOTView<x>:CURSor:SCREEN:BYPosition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:SCREEN:BYPosition <NR3> DISplay:PLOTView<x>:CURSor:SCREEN:BYPosition?
Arguments	PLOTView<x> is the Plot waveform number. <NR3> is the vertical cursor B position.
Examples	DISplay:PLOTView1:CURSor:SCREEN:BYPosition 0.589 sets the B cursor Y position to 589 mV.

DISplay:PLOTView1:CURSor:SCREEN:BYPOSITION? might return
:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BYPOSITION 589.0696E-3
indicating the B cursor Y position is 589.0696 mV.

DISplay:PLOTView<x>:CURSor:SPLITMODE

This command sets or queries the cursor source mode in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:SPLITMODE {SAME SPLIT} DISplay:PLOTView<x>:CURSor:SPLITMODE?
Arguments	PLOTView<x> is the Plot waveform number. SAME specifies that both cursors are on the same waveform. SPLIT specifies that the cursors can be on different waveforms.
Examples	DISplay:PLOTView1:CURSor:SPLITMODE SAME sets the cursors to be on the same waveform. DISplay:PLOTView1:CURSor:SPLITMODE? might return :DISPLAY:PLOTVIEW1:CURSOR:SPLITMODE SAME indicating both cursors are on the same waveform.

DISplay:PLOTView<x>:CURSor:STATE

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:STATE {OFF ON <NR1>} DISplay:PLOTView<x>:CURSor:STATE?
Arguments	PLOTView<x> is the Plot waveform number. OFF disables the specified cursor. ON enables the specified cursor.

<NR1> = 0 disables the specified cursor; any other value enables the specified cursor.

Examples `DISPlay:PLOTView1:CURSor:STATE OFF` disables the specified cursor.
`DISPlay:PLOTView5:CURSor:STATE?` might return
`:DISPLAY:PLOTVIEW5:CURSOR:STATE 1` indicating cursors are on.

DISPlay:PLOTView<x>:CURSor:VBArS:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSor:VBArS:APOSition <NR3>`
`DISPlay:PLOTView<x>:CURSor:VBArS:APOSition?`

Arguments `PLOTView<x>` is the Plot waveform number.
`<NR3>` is the vertical cursor A position.

Examples `DISPlay:PLOTView1:CURSor:VBArS:APOSition 50.0E-9` sets the VBARS APOSITION to 50 ns.
`DISPlay:PLOTView1:CURSor:VBArS:APOSition?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:VBARS:APOSITION -60.0796E-9`
indicating the VBARS APOSITION is set to -60.08 ns.

DISPlay:PLOTView<x>:CURSor:VBArS:BPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSor:VBArS:BPOSition <NR3>`
`DISPlay:PLOTView<x>:CURSor:VBArS:BPOSition?`

Arguments	PLOTView<x> is the Plot waveform number. <NR3> is the vertical cursor B position.
Examples	:DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION 50.0E-9 sets the VBARS BPOSITION to 50 ns. :DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION -60.0796E-9 indicating the VBARS BPOSITION is set to 60.08 ns.

DISplay:PLOTView<x>:CURSOR:VBARS:DELTA? (Query Only)

This command queries the delta T cursor readout value of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSOR:VBARS:DELTA?
Arguments	PLOTView<x> is the Plot waveform number.
Returns	The delta T cursor readout value is returned.
Examples	DISplay:PLOTView1:CURSOR:VBARS:DELTA? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:DELTA 119.7593E-9 indicating the VBARS DELTA is set to 119.76 ns.

DISplay:PLOTView<x>:CURSOR:VBARS:UNIts? (Query Only)

This command queries the VBARS cursor readout units of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSOR:VBARS:UNIts?
Arguments	PLOTView<x> is the Plot waveform number.

Returns The VBArS cursor readout units are returned.

Examples `DISPlay:PLOTView<x>:CURSOR:VBArS:UNITs?` might return
`:DISPLAY:PLOTVIEW5:CURSOR:VBARS:UNITS "V"` indicating the units are
Volts.

DISPlay:PLOTView<x>:CURSOR:WAVEform:APOSITION

This command sets or queries the waveform cursor A horizontal position of the specified cursor in the specified view.

NOTE. *In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command sets or returns the cursor A vertical position. For all other plots, this command sets or returns the cursor A horizontal position.*

Group Cursor

Syntax `DISPlay:PLOTView<x>:CURSOR:WAVEform:APOSITION <NR3>`
`DISPlay:PLOTView<x>:CURSOR:WAVEform:APOSITION?`

Arguments `PLOTView<x>` is the Plot waveform number.
`<NR3>` is the horizontal cursor A position.

Examples `DISPlay:PLOTView1:CURSOR:WAVEform:APOSITION -50.0E-9` sets the
waveform APOSITION to -50.0 ns.
`DISPlay:PLOTView1:CURSOR:WAVEform:APOSITION?` might return
`:DISPLAY:PLOTVIEW1:CURSOR:WAVEFORM:APOSITION -60.0796E-9`
indicating the waveform APOSITION is set to -60.08 ns.

DISPlay:PLOTView<x>:CURSOR:WAVEform:BPOSITION

This command sets or queries the waveform cursor B horizontal position of the specified cursor in the specified view.

NOTE. *In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command has no effect. Query returns invalid values. For all other plots, this command sets or returns the cursor B horizontal position.*

Group	Cursor
Syntax	DISplay:PLOTView<x>:CURSor:WAVEform:BPOsition <NR3> DISplay:PLOTView<x>:CURSor:WAVEform:BPOsition?
Arguments	PLOTView<x> is the Plot waveform number. <NR3> is the horizontal cursor B position.
Examples	DISplay:PLOTView1:CURSor:WAVEform:BPOsition 3.0 sets the position to 3.0 V. DISplay:PLOTView5:CURSor:WAVEform:BPOsition? might return :DISPLAY:PLOTVIEW5:CURSOR:WAVEFORM:BPOSITION 2.600517840 indicating the position is 2.5 V.

DISplay:PLOTView<x>:GRIDlines

This command sets or queries the Grid (graticule) lines setting of the specified plot. This command works for plots that have vertical and horizontal units associated with the graticule. For example, this command does not work for XY or XYZ plots.

Group	Display Control
Syntax	DISplay:PLOTView<x>:GRIDlines {HORizontal VERTical BOTH} DISplay:PLOTView<x>:GRIDlines?
Arguments	PLOTView<x> is the Plot waveform number. HORizontal specifies horizontal grid lines. VERTical specifies vertical grid lines. BOTH specifies both vertical and horizontal grid lines.
Examples	DISplay:PLOTView1:GRIDlines vertical sets the vertical grid lines to display. DISplay:PLOTView1:GRIDlines? might return :DISPLAY:PLOTVIEW1:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are displayed.

DISPLAY:PLOTVIEW<x>:XAXIS:SCALE

This command sets or queries the horizontal scale setting for applicable plots (Linear or Log) for the specified plot view.

Group	Display
Syntax	DISPLAY:PLOTVIEW<x>:XAXIS:SCALE {LINEAR LOG}
Arguments	<p>PLOTVIEW<x> is the Plot waveform number.</p> <p>LINEAR creates a plot with linear scales.</p> <p>LOG creates a plot with logarithmic scales.</p>
Examples	<p>DISPLAY:PLOTVIEW1:XAXIS:SCALE LOG creates a plot with a logarithmic horizontal scale.</p> <p>DISPLAY:PLOTVIEW1:XAXIS:SCALE? might return :DISPLAY:PLOTVIEW1:XAXIS:SCALE LINEAR indicating plots will have linear scales.</p>

DISPLAY:PLOTVIEW<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting for applicable plots (Linear or Log) in the specified plot view.

Group	Display
Syntax	DISPLAY:PLOTVIEW<x>:YAXIS:SCALE {LINEAR LOG}
Arguments	<p>PLOTVIEW<x> is the Plot waveform number.</p> <p>LINEAR specifies a linear vertical scale.</p> <p>LOG specifies a logarithmic vertical scale.</p>
Examples	<p>DISPLAY:PLOTVIEW1:YAXIS:SCALE LINEAR sets the plot to use a linear vertical scale.</p> <p>DISPLAY:PLOTVIEW1:YAXIS:SCALE? might return :DISPLAY:PLOTVIEW1:YAXIS:SCALE LOG indicating a logarithmic vertical scale is used.</p>

DISplay:PLOTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the specified plot.

Group	Zoom
Syntax	DISplay:PLOTView<x>:ZOOM:XAXIS:FROM <NR3> DISplay:PLOTView<x>:ZOOM:XAXIS:FROM?
Arguments	PLOTView<x> is the Plot waveform number. <NR3> is start of the zoom x-axis.
Examples	DISplay:PLOTView1:ZOOM:XAXIS:FROM 2.0 sets the start of the zoom x-axis to 2.0 V. DISplay:PLOTView3:ZOOM:XAXIS:FROM? might return DISPLAY:PLOTVIEW3:ZOOM:XAXIS:FROM 1.9844803459459 indicating the start of the zoom x-axis is 1.9845 V.

DISplay:PLOTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge of the specified plot.

Group	Zoom
Syntax	DISplay:PLOTView<x>:ZOOM:XAXIS:TO <NR3> DISplay:PLOTView<x>:ZOOM:XAXIS:TO?
Arguments	PLOTView<x> is the Plot waveform number. <NR3> is the end of the zoom x-axis.
Examples	DISplay:PLOTView1:ZOOM:XAXIS:TO 2.5 sets the end of the zoom x-axis to 2.5 V. DISplay:PLOTView3:ZOOM:XAXIS:TO? might return DISPLAY:PLOTVIEW3:ZOOM:XAXIS:TO 2.2144883507508 indicating the end of the zoom x-axis is 2.21 V.

DISplay:PLOTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax DISplay:PLOTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM?

Arguments PLOTView<x> is the Plot waveform number.
<NR3> is the bottom value of the zoom y-axis.

Examples DISplay:PLOTView1:ZOOM:YAXIS:FROM -1.0E+20 sets the bottom value of the zoom y-axis to -1E+20.

DISplay:PLOTView1:ZOOM:YAXIS:FROM? might return
:DISPLAY:PLOTVIEW1:ZOOM:YAXIS:FROM -1.0E+21 indicating
the bottom value of the zoom y-axis is -1E+21.

DISplay:PLOTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax DISplay:PLOTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:TO?

Arguments PLOTView<x> is the Plot waveform number.
<NR3> is the top value of the zoom y-axis.

Examples DISplay:PLOTView1:ZOOM:YAXIS:TO 100 sets the top value of the zoom y-axis to 100.

DISplay:PLOTView1:ZOOM:YAXIS:TO? might return
:DISPLAY:PLOTVIEW3:ZOOM:YAXIS:FROM 0.0E+0 indicating the
top value of the zoom y-axis is 0.0 hits (for a histogram plot, or it could be Volts
for an XY plot).

DISplay:REFFFTView<n>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group	Cursor
Syntax	DISplay:REFFFTView<n>:CURSor:ROLOCATION {GRATICULE BADGE} DISplay:REFFFTView<n>:CURSor:ROLOCATION?
Arguments	<p>PLOTView<x> is the Reference FFT plot number.</p> <p>GRATICULE sets the Reference FFT plot cursor readouts to display as part of the cursors in the plot view.</p> <p>BADGE removes the Reference FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.</p>
Examples	<p>DISplay:REFFFTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for Reference FFT plot 1 to display in the plot graticule.</p> <p>DISplay:REFFFTView2:CURSor:ROLOCATION? might return DISplay:REFFFTView2:CURSor:ROLOCATION BADGE, indicating that cursor readout information for Reference FFT plot 2 is set to display as a badge in the Results Bar.</p>

DISplay:REFFFTView<x>:AUTOScale

This command sets or queries the enabled state of auto-scale for plots.

Group	Display Control
Syntax	DISplay:REFFFTView<x>:AUTOScale {OFF ON 0 1} DISplay:REFFFTView<x>:AUTOScale?
Arguments	<p>REFFFTView<x> is the plot number.</p> <p><NR1> = 0 disables auto-scale on the specified reffftview; any other value turns this feature on.</p> <p>OFF disables auto-scale on the specified reffftview.</p> <p>ON enables the specified channel on the specified Waveform View.</p>

Examples `DISPlay:REFFFTView1:AUTOScale 1` enables auto-scale on the specified view.

`DISPlay:REFFFTView5:AUTOScale?` might return
`:DISPLAY:REFFFTVIEW5:AUTOSCALE 1` indicating auto-scale is on.

DISPlay:REFFFTView<x>:CURSor:ASOURce? (Query Only)

This command returns the cursor source for plot cursor A

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:ASOURce?`

Arguments `PLOTView<x>` is the Plot waveform number.

Returns Returns the cursor source for plot cursor A.

Examples `DISPlay:REFFFTView5:CURSor:ASOURce?` might return
`:DISPLAY:REFFFTVIEW5:CURSOR:ASOURCE REF5` indicating the A cursor source is reference 5.

DISPlay:REFFFTView<x>:CURSor:BSOURce? (Query Only)

This command returns the cursor source for plot cursor B.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:BSOURce?`

Returns Returns the cursor source for plot cursor B.

Examples `DISPlay:REFFFTView5:CURSor:BSOURce?` might return
`:DISPLAY:REFFFTVIEW5:CURSOR:BSOURCE REF5` indicating the source of the B cursor is reference 5.

DISplay:REFFFTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTview<x>:CURSor:DDT?
Returns	Returns the delta V over delta T cursor readout value
Examples	DISplay:REFFFTview5:CURSor:DDT? might return :DISPLAY:REFFFTVIEW5:CURSOR:DDT 9.91E+37 indicating the delta V over delta T cursor readout value is 9.91E+37.

DISplay:REFFFTView<x>:CURSor:FUNcTion

This command sets or queries the cursor type of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTview<x>:CURSor:FUNcTion {WAVEform VBArS HBArS SCREEN} DISplay:REFFFTview<x>:CURSor:FUNcTion?
Arguments	<p>HBArS specifies horizontal bar cursors, which measure in vertical units.</p> <p>VBArS specifies vertical bar cursors, which measure in horizontal units.</p> <p>SCREEN specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the cursor sources. Use these cursors to measure anywhere in the waveform display area.</p> <p>WAVEform specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.</p>
Examples	DISplay:REFFFTview5:CURSor:FUNcTion screen sets the cursor function to screen.

DISplay:REFFFTView5:CURSor:FUNction? might return
:DISPLAY:REFFFTVIEW5:CURSOR:FUNCTION WAVEFORM indicating the cursor
function is set to waveform.

DISplay:REFFFTView<x>:CURSor:HBArS:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:HBArS:APOSition <NR3> DISplay:REFFFTView<x>:CURSor:HBArS:APOSition?
Arguments	<NR3> is the vertical cursor A position of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:HBArS:APOSition -20 sets the A cursor position to -20. DISplay:REFFFTView5:CURSor:HBArS:APOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:APOSITION -14.4762489421361 indicating the A cursor position is -14.48 dBm.

DISplay:REFFFTView<x>:CURSor:HBArS:AUNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:HBArS:AUNIts?
Returns	Returns the cursor A vertical units of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:HBArS:AUNIts? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:AUNITS "dBm" indicating the A cursor units are dBm.

DISplay:REFFFTView<x>:CURSor:HBARs:BPOStion

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:HBARs:BPOStion <NR3> DISplay:REFFFTView<x>:CURSor:HBARs:BPOStion?
Arguments	<NR3> is the vertical cursor B position of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:HBARs:BPOStion -90 sets the B cursor position to -90. DISplay:REFFFTView5:CURSor:HBARs:BPOStion? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BPOSITION -100.6262359333038 indicating the B cursor position is -100.6.

DISplay:REFFFTView<x>:CURSor:HBARs:BUNIts? (Query Only)

This command returns the cursor B vertical units of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:HBARs:BUNIts?
Returns	Returns the cursor B vertical units of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:HBARs:BUNIts? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BUNITS "dBm" indicating the units are dBm.

DISplay:REFFFTView<x>:CURSor:HBARs:DELTA? (Query Only)

This command returns the delta V cursor readout value of the specified cursor in the specified view.

Group	Cursor
Syntax	<code>DISPlay:REFFFTView<x>:CURSor:HBARS:DELTA?</code>
Returns	Returns the delta V cursor readout value of the specified cursor in the specified view.
Examples	<code>DISPlay:REFFFTView5:CURSor:HBARS:DELTA?</code> might return <code>:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:DELTA 59.9867396737569</code> indicating the delta V cursor readout value is 59.99.

`DISPlay:REFFFTView<x>:CURSor:MODE`

This command sets or queries the cursor tracking mode of the specified cursor in the specified view.

Group	Cursor
Syntax	<code>DISPlay:REFFFTView<x>:CURSor:MODE {INDEPENDENT TRACK}</code> <code>DISPlay:REFFFTView<x>:CURSor:MODE?</code>
Arguments	<code>TRACK</code> ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A. <code>INDEPENDENT</code> allows independent adjustment of the two cursors.
Examples	<code>DISPlay:REFFFTView5:CURSor:MODE TRACK</code> sets the cursors to track together. <code>DISPlay:REFFFTView5:CURSor:MODE?</code> might return <code>:DISPLAY:REFFFTVIEW5:CURSOR:MODE INDEPENDENT</code> indicating the cursors move independently.

`DISPlay:REFFFTView<x>:CURSor:ONEOVERDELTAVALUE? (Query Only)`

This command returns the one over delta T cursor readout value of the specified cursor in the specified view.

Group	Cursor
--------------	--------

Syntax `DISPlay:REFFFTView<x>:CURSor:ONEOVERDELTAVALUE?`

Examples `DISPlay:REFFFTView5:CURSor:ONEOVERDELTAVALUE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:ONEOVERDELTAVALUE 2.540935140340E-9` indicating the one over delta T cursor readout value is 2.54 ns.

DISPlay:REFFFTView<x>:CURSor:SCREEN:AXPOSITION

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:SCREEN:AXPOSITION <NR3>`
`DISPlay:REFFFTView<x>:CURSor:SCREEN:AXPOSITION?`

Arguments `<NR3>` is the horizontal cursor A position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSor:SCREEN:AXPOSITION 1.0E+6` sets the cursor position to 1.0 MHz.

`DISPlay:REFFFTView5:CURSor:SCREEN:AXPOSITION?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AXPOSITION 792.4465962305570E+3` indicating the A cursor horizontal position is 792.447 kHz.

DISPlay:REFFFTView<x>:CURSor:SCREEN:AYPOSITION

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:SCREEN:AYPOSITION <NR3>`
`DISPlay:REFFFTView<x>:CURSor:SCREEN:AYPOSITION?`

Arguments `<NR3>` is the vertical cursor A position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSor:SCREEN:AYPosition -20.9` sets the position to -20 dBm.

`DISPlay:REFFFTView5:CURSor:SCREEN:AYPosition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AYPOSITION -30.0132603262431` indicating the A cursor vertical position is -30.01 dBm.

DISPlay:REFFFTView<x>:CURSor:SCREEN:BXPosition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:SCREEN:BXPosition <NR3>`
`DISPlay:REFFFTView<x>:CURSor:SCREEN:BXPosition?`

Arguments <NR3> is the horizontal cursor B position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSor:SCREEN:BXPosition 300` sets the position to 300 MHz.

`DISPlay:REFFFTView5:CURSor:SCREEN:BXPosition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BXPOSITION 394.3483403001212E+6` indicating the B cursor horizontal position is 394.348 MHz.

DISPlay:REFFFTView<x>:CURSor:SCREEN:BYPosition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:SCREEN:BYPosition <NR3>`
`DISPlay:REFFFTView<x>:CURSor:SCREEN:BYPosition?`

Arguments <NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSOR:SCREEN:BYPOSITION -80` sets the cursor position to -90 dBm.

`DISPlay:REFFFTView5:CURSOR:SCREEN:BYPOSITION?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BYPOSITION -90.0` indicating the B cursor vertical position is -90 dBm.

DISPlay:REFFFTView<x>:CURSOR:SPLITMODE

This command sets or queries whether both cursors have same or different source.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSOR:SPLITMODE {SAME|SPLIT}`
`DISPlay:REFFFTView<x>:CURSOR:SPLITMODE?`

Arguments SAME specifies both cursors have the same sources.
 SPLIT specifies both cursors have different sources.

Examples `DISPlay:REFFFTView5:CURSOR:SPLITMODE SPLIT` specified that the cursors have different sources.

`DISPlay:REFFFTView5:CURSOR:SPLITMODE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SPLITMODE SAME` indicating both cursors have the same source.

DISPlay:REFFFTView<x>:CURSOR:STATE

This command sets or queries the visible state of the cursor of the specified cursor n the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSOR:STATE {OFF|ON|0|1}`
`DISPlay:REFFFTView<x>:CURSOR:STATE?`

Arguments <NR1> = 0 specifies the cursor is not visible; any other value displays the cursor.
 OFF specifies the cursor is not visible.
 ON displays the cursor.

Examples `DISPlay:REFFFTView5:CURSor:STATE ON` specifies the cursor is visible.
`DISPlay:REFFFTView5:CURSor:STATE?` might return
`:DISPLAY:REFFFTVIEW5:CURSOR:STATE 1` indicating the cursor is visible.

DISPlay:REFFFTView<x>:CURSor:VBArS:APOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:VBArS:APOSition <NR3>`
`DISPlay:REFFFTView<x>:CURSor:VBArS:APOSition?`

Arguments <NR3> is the horizontal cursor A position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSor:VBArS:APOSition 1.0E+6` sets the cursor to 1 MHz.
`DISPlay:REFFFTView5:CURSor:VBArS:APOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:VBARS:APOSITION 792.4465962305570E+3` indicating the cursor position is 792.447 kHz.

DISPlay:REFFFTView<x>:CURSor:VBArS:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax `DISPlay:REFFFTView<x>:CURSor:VBArS:BPOSition <NR3>`
`DISPlay:REFFFTView<x>:CURSor:VBArS:BPOSition?`

Arguments <NR3> is the horizontal cursor B position of the specified cursor in the specified view.

Examples `DISPlay:REFFFTView5:CURSor:VBArS:BPOSition 300.0E+6` sets the cursor position to 300 MHz.

```
DISplay:REFFFTView5:CURSor:VBARS:BPOSITION? might  
return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:BPOSITION  
394.3483403001212E+6 indicating the B cursor horizontal position is  
394.348 MHz.
```

DISplay:REFFFTView<x>:CURSor:VBARS:DELTA? (Query Only)

This command returns the delta T cursor readout value of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:VBARS:DELTA?
Returns	Returns the delta T cursor readout value of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:VBARS:DELTA? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:DELTA 393.5558937038906E+6 indicating the delta T cursor readout value is 393.556 MHz.

DISplay:REFFFTView<x>:CURSor:VBARS:UNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:VBARS:UNIts?
Returns	Returns cursor A vertical units of the specified cursor in the specified view.
Examples	DISplay:REFFFTView5:CURSor:VBARS:UNIts? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:UNITS "Hz" indicating the A cursor vertical units are Hz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOsition? (Query Only)

This query-only command returns the value of the cursor A horizontal position.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOsition?
Examples	DISplay:REFFFTView1:CURSor:WAVEform:AHPOsition? might return :DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:AHPOSITION 9.91E+37.

DISplay:REFFFTView<x>:CURSor:WAVEform:APOsition

Sets or returns the waveform cursor A position in the specified plot view.

NOTE. *In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command sets or queries the cursor A vertical position. For all other plots, this command sets or queries the cursor A horizontal position.*

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:WAVEform:APOsition <NR3> DISplay:REFFFTView<x>:CURSor:WAVEform:APOsition?
Arguments	<NR3> is the waveform cursor A position in the specified plot view.
Examples	DISplay:REFFFTView5:CURSor:WAVEform:APOsition 7E+5 sets the position to 700 kHz. DISplay:REFFFTView5:CURSor:WAVEform:APOsition? might return :DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:APOSITION 792.4465962305570E+3 indicating the A cursor waveform position is 792.4 kHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOsition? (Query Only)

This query-only command returns the value of the cursor A vertical position.

Group	Cursor
Syntax	DISPlay:REFFFTView<x>:CURSor:WAVEform:AVPOSITION?
Examples	DISPlay:REFFFTView<x>:CURSor:WAVEform:AVPOSITION? might return :DISPlay:REFFFTView1:CURSor:WAVEform:AVPOSITION 1.605E+0 indicating the cursor vertical position is 1.6 dBm.

DISPlay:REFFFTView<x>:CURSor:WAVEform:BHPOSITION? (Query Only)

This query-only command returns the value of the cursor B horizontal position.

Group	Cursor
Syntax	DISPlay:REFFFTView<x>:CURSor:WAVEform:BHPOSITION?
Examples	DISPlay:REFFFTView<x>:CURSor:WAVEform:BHPOSITION? might return :DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:BHPOSITION 9.91E+37.

DISPlay:REFFFTView<x>:CURSor:WAVEform:BPOSITION

Sets or returns the waveform cursor B position in the specified plot view.

NOTE. In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command has no effect. Queries return invalid values. For all other plots, this command sets or queries the cursor B horizontal position.

Group	Cursor
Syntax	DISPlay:REFFFTView<x>:CURSor:WAVEform:BPOSITION <NR3> DISPlay:REFFFTView<x>:CURSor:WAVEform:BPOSITION?
Arguments	<NR3> is the waveform cursor B position in the specified plot view.
Examples	DISPlay:REFFFTView5:CURSor:WAVEform:BPOSITION 4E+8 sets the cursor position to 400 MHz.

DISplay:REFFFTView5:CURSor:WAVEform:BPOsition? might
return :DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:BPOSITION
394.3483403001212E+6 indicating the B waveform cursor position is
394.35 MHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOsition? (Query Only)

This query-only command returns the value of the cursor B vertical position.

Group	Cursor
Syntax	DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOsition?
Examples	DISplay:REFFFTView1:CURSor:WAVEform:BVPOsition? might return :DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOsition -119.866E+0 indicating the cursor vertical position is -119.9 dBm.

DISplay:REFFFTView<x>:GRIDLines

This command sets or returns the grid lines setting of the plot.

Group	Display Control
Syntax	DISplay:REFFFTView<x>:GRIDLines {HORizontal VERTical BOTH} DISplay:REFFFTView<x>:GRIDLines?
Arguments	HORizontal specifies horizontal grid lines. VERTical specifies vertical grid lines. BOTH specifies both horizontal and vertical grid lines.
Examples	DISplay:REFFFTView5:GRIDLines hhorizontal specifies horizontal grid lines. DISplay:REFFFTView5:GRIDLines? might return :DISPLAY:REFFFTVIEW5:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are displayed

DISplay:REFFFTView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

Group Display Control

Syntax DISplay:REFFFTView<x>:REF:REF<x>:STATE *boolean*
DISplay:REFFFTView<x>:REF:REF<x>:STATE?

Arguments <NR1> = 0 disables the specified reference; any other value turns this feature on.
OFF disables the display the specified reference.
ON enables the specified reference.

Examples DISplay:REFFFTView5:REF:REF5:STATE 0 turns display of the reference off.
DISplay:REFFFTView5:REF:REF5:STATE? might return
:DISPLAY:REFFFTVIEW5:REF:REF5:STATE 1 indicating the reference is displayed.

DISplay:REFFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale setting for Ref FFT.

Group Display Control

Syntax DISplay:REFFFTView<x>:XAXIS:SCALE {LINEAR|LOG}
DISplay:REFFFTView<x>:XAXIS:SCALE?

Arguments LINEAR specifies a linear scale.
LOG specifies a logarithmic scale.

Examples DISplay:REFFFTView5:XAXIS:SCALE linear sets the x-axis scale to linear.
DISplay:REFFFTView5:XAXIS:SCALE? might return
:DISPLAY:REFFFTVIEW5:XAXIS:SCALE LOG indicating x-axis scale for the specified reference view is logarithmic.

DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM

This command sets or returns the left edge of the zoom x-axis in the specified plot view.

Group Zoom

Syntax DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM?

Arguments <NR3> is the left edge of the zoom x-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:XAXIS:FROM 2.0E+6 sets the left edge of the specified view to 2.0 MHz.

DISplay:REFFFTView5:ZOOM:XAXIS:FROM? might return
:DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:FROM 1.3295739742362E+6
indicating the left edge of the specified view is 1.33 MHz.

DISplay:REFFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the right edge of the zoom x-axis in the specified plot view.

Group Zoom

Syntax DISplay:REFFFTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO?

Arguments <NR3> is the right edge of the zoom x-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:XAXIS:TO 2.0E+8 sets the right edge of the zoom x-axis to 200 MHz.

DISplay:REFFFTView5:ZOOM:XAXIS:TO? might return
:DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:TO 235.0376933178995E+6
indicating the right edge of the zoom x-axis is 235.0 MHz.

DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group	Zoom
Syntax	DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM <NR3> DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM?
Arguments	<NR3> is the bottom value of the zoom y-axis in the specified plot view.
Examples	DISplay:REFFFTView5:ZOOM:YAXIS:FROM -120.0 sets the bottom value of the zoom y-axis to -120.0 dBm. DISplay:REFFFTView5:ZOOM:YAXIS:FROM? might return :DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:FROM -129.7533120632172 indicating the bottom value of the zoom y-axis is -129.8 dBm.

DISplay:REFFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group	Zoom
Syntax	DISplay:REFFFTView<x>:ZOOM:YAXIS:TO <NR3> DISplay:REFFFTView<x>:ZOOM:YAXIS:TO?
Arguments	<NR3> is the top value of the zoom y-axis in the specified plot view.
Examples	DISplay:REFFFTView5:ZOOM:YAXIS:TO 0.0 sets the top value of the zoom y-axis to 0.0 dBm. DISplay:REFFFTView5:ZOOM:YAXIS:TO? might return :DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:TO 15.8820685863495 indicating the top value of the zoom y-axis is 15.9 dBm.

DISplay:SElect:BUS

This command sets or queries the overall selected bus. Sets are applied to all views that contain the source and the selected view is changed. When multiple buses are open, querying the command gives the correct result, but the bus cannot set.

Group Display Control

Syntax DISplay:SElect:BUS BUS<x>
DISplay:SElect:BUS?

Arguments BUS<x> is the selected bus.

Examples DISplay:SElect:BUS Bus2 sets Bus 2 as the selected bus.

DISplay:SElect:BUS? might return :DISPLAY:SELECT:BUS BUS1
indicating Bus 1 is the selected bus.

DISplay:SElect:MATH

This command sets or queries the overall selected math. Sets are applied to all views that contain the source and the selected view is changed. When multiple Math are open, querying the command gives the correct result, but the required Math cannot be set.

Group Display Control

Syntax DISplay:SElect:MATH MATH<x>
DISplay:SElect:MATH?

Arguments MATH<x> is the selected math.

Examples DISplay:SElect:MATH MATH1 selects Math 1.

DISplay:SElect:MATH? might display :DISPLAY:SELECT:MATH MATH2
indicating that Math 2 is selected.

DISplay:SElect:REFerence

This command sets or queries the overall selected reference waveform. Sets are applied to all views that contain the source and the selected view is changed.

Group	Display Control
Syntax	DISplay:SElect:REFerence {NONE REF<x>}
Arguments	Arguments are the selected reference.
Examples	DISplay:SElect:REFerence REF2 selects reference 2. DISplay:SElect:REFERENCE? might return DISPLAY:SELECT:REFERENCE NONE indicating no reference is selected.

DISplay:SElect:SOUrce

This command sets or queries the overall selected source. Sets are applied to all views that contain the source and the selected view is changed.

Group	Display Control
Syntax	DISplay:SElect:SOUrce {NONE CH<x> DCH<x> BUS<x> MATH<x> PLOT<x> REF<x>}
Arguments	NONE disables the selected source. CH<x> specifies an analog channel to use as the source. DCH<x> specifies a digital channel to use as the source. The supported digital channel value is 1. MATH<x> specifies a math waveform to use as the source. REF<x> specifies a reference waveform to use as the source. PLOT<x> specifies a plot as the source.
Examples	DISplay:SElect:SOUrce Bus1 selects Bus1 as the selected source. DISplay:SElect:SOUrce? might return :DISPLAY:SELECT:SOUrce MATH1 indicating Math 1 is the selected source.

DISplay:SElect:VIEW

This command sets or queries the selected view.

Group Display Control

Syntax DISplay:SElect:VIEW
{WAVEVIEW1|MATHFFT<x>|PLOTVIEW<x>|REFFFT<x>}

Arguments Arguments are the selected view.

Examples DISplay:SElect:VIEW PLOTVIEW1 selects PLOTVIEW1 as the selected view.
DISplay:SElect:VIEW? might return :DISPLAY:SELECT:VIEW MATHFFT1 indicating MATHFFT1 is the selected view.

DISplay:SElect:WAVEView<x>:SOURce

This command sets or queries the selected source in the given waveview.

Group Display Control

Syntax DISplay:SElect:WAVEView<x>:SOURce
{CH<x>|MATH<x>|BUS<x>|REF<x>|PLOT<x>}

Arguments Arguments are the selected source.

Examples DISplay:SElect:WAVEView1:SOURce MATH1 sets MATH1 as the selected source.
DISplay:SElect:WAVEView1:SOURce? might return :DISPLAY:SELECT:WAVEVIEW1:SOURCE CH2 indicating CH2 is the selected source.

DISplay:VARpersist

This command sets or queries display persistence decay time, which is the approximate decay time for a freshly struck persistence sample.

Group Display Control

Syntax `DISplay:VARpersist <NR3>`
`DISplay:VARpersist?`

Related Commands [DISplay:PERSistence](#)

Arguments <NR3> indicates the persistence decay time and ranges from 0.5 to 100.

Examples `DISPLAY:VARPERSIST 5` sets the persistence decay time to 5.
`DISPLAY:VARPERSIST?` might return `:DISPLAY:VARPERSIST 3.0000E-01`, indicating that persistence decay time is currently set to 0.300.

DISplay:WAVEform

This command globally enables or disables the waveform display. When disabled, the waveform is still acquired and held in memory, but it is not drawn to the screen. Disabling the waveform display may improve processing speed.

Group Cursor

Syntax `DISplay:WAVEform {<NR1>|OFF|ON}`
`DISplay:WAVEform?`

Arguments <NR1> enables or disables the waveform display. 0 disables the waveform display; any other value enables the waveform display.

ON enables the waveform display.

OFF disables the waveform display.

Examples `DISplay:WAVEform OFF` disables the waveform display.
`DISplay:WAVEform?` might return `DISplay:WAVEform 1`, indicating that the waveform display is enabled.

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION

This command sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).

Group Cursor

Syntax `DISPlay:WAVEView:CURSor:CURSOR1:ROLOCATION {GRATICULE|BADGE}`
`DISPlay:WAVEView:CURSor:CURSOR1:ROLOCATION?`

Arguments `GRATICULE` sets the Waveform View cursor readouts to display as part of the cursors in the plot view.

`BADGE` removes the Waveform View cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples `DISPlay:WAVEView:CURSor:CURSOR1:ROLOCATION GRATICULE` sets the cursor readout information for the Waveform View to display in the Waveform View graticule.

`DISPlay:WAVEView:CURSor:CURSOR1:ROLOCATION?` might return `DISPlay:WAVEView:CURSor:CURSOR1:ROLOCATION BADGE`, indicating that cursor readout information for the Waveform View is set to display as a badge in the Results Bar.

DISPlay:WAVEView:GRIDTYPE

This command sets or queries the Waveform View Graticule type.

Group Display

Syntax `DISPlay:WAVEView:GRIDTYPE {MOVEABLE|FIXED}`
`DISPlay:WAVEView:GRIDTYPE?`

Arguments `MOVEABLE` sets the Waveform View so that both the waveform and the grid (graticule) move together when moving the waveform horizontally

`FIXED` sets the Waveform View so that the grid does not move when moving the waveform horizontally.

Examples `DISPlay:WAVEView:GRIDTYPE MOVEABLE` sets the Waveform View to enable both the graticule and the waveform to move together.

`DISPlay:WAVEView:GRIDTYPE?` might return `FIXED`, indicating that the graticule will not change positions if the waveform moves horizontally.

DISPlay:WAVEView<x>:BUS:B<x>:STATE

Sets or queries the state of the specified bus in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISPlay:WAVEView<x>:BUS:B<x>:STATE {OFF ON 0 1} DISPlay:WAVEView<x>:BUS:B<x>:STATE?
Arguments	0 turns specified bus off. 1 turns the specified bus on. ON turns the specified bus on. OFF turns specified bus off.
Examples	DISPlay:WAVEView1:BUS:B1:STATE OFF turns specified bus off. DISPlay:WAVEView1:BUS:B1:STATE? might return :DISPLAY:WAVEVIEW1:BUS:B1:STATE 1 indicating the specified bus is on.

DISPlay:WAVEView<x>:BUS:B<x>:VERTical:POSition

Sets or queries the vertical position of the specified bus in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISPlay:WAVEView<x>:BUS:B<x>:VERTical:POSition <NR3> DISPlay:WAVEView<x>:BUS:B<x>:VERTical:POSition?
Arguments	<NR3> is the vertical position of the specified bus.
Examples	DISPlay:WAVEView1:BUS:B1:VERTical:POSition 1.0e0 sets the position to 1 division. DISPlay:WAVEView1:BUS:B1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:BUS:B1:VERTICAL:POSITION 0.0E+0 indicating the position is 0.0 divisions.

DISplay:WAVEView<x>:CH<x>:STATE

Sets or queries the state of the specified channel in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.

Group Display Control

Syntax DISplay:WAVEView<x>:CH<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CH<x>:STATE?

Arguments <NR1> = 0 disables the specified channel on the specified Waveform View; any other value turns this feature on.

OFF disables the display the specified channel on the specified Waveform View.

ON enables the specified channel on the specified Waveform View.

Examples DISplay:WAVEView1:CH1:STATE ON turns on channel 1.

DISplay:WAVEView1:CH1:STATE? might return
:DISPLAY:WAVEVIEW1:CH1:STATE 0 indicating channel 1 is off.

DISplay:WAVEView<x>:CH<x>:VERTical:POSition

Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.

Group Display Control

Syntax DISplay:WAVEView<x>:CH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:POSition?

Arguments <NR3> is the vertical position in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

Examples `DISPlay:WAVEView1:CH1:VERTical:POSition 1.0e0` sets the position to 1 division.

`DISPlay:WAVEView1:CH1:VERTical:POSition?` might return `:DISPLAY:WAVEVIEW1:CH1:VERTICAL:POSITION -3.4400` indicating the channel 1 vertical position is -3.44 divisions.

DISplay:WAVEView<x>:CH<x>:VERTical:SCALE

Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax `DISPlay:WAVEView<x>:CH<x>:VERTical:SCALE <NR3>`
`DISPlay:WAVEView<x>:CH<x>:VERTical:SCALE?`

Arguments <NR3> is the vertical scale of the specified channel.

Examples `DISPlay:WAVEView1:CH1:VERTical:SCALE 2.0e2` sets the vertical scale to 200 mV per division.

`DISPlay:WAVEView1:CH1:VERTical:SCALE?` might return `:DISPLAY:WAVEVIEW1:CH1:VERTICAL:SCALE 200.0000E-3` indicating the scale is 200 mV per division.

DISplay:WAVEView<x>:CURSor? (Query Only)

This query returns the cursor parameters for the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Cursor

Syntax `DISPlay:WAVEView<x>:CURSor?`

Returns Returns the cursor parameters for the specified Waveform View.

Examples DISPlay:WAVEView1:CURSor? might return
 WAVEFORM;AUTO;2.4993299784074;2.4958885115371;
 -3.4414668702805E-3;"A";"A";300.0E-6;2.2369669999970;
 256.967000E-3;-300.0E-6;-300.0E-6;300.0E-6; "????
 ?????";"s";"???? ????";600.0E-6;1.6666666666667E+3;
 1;300.0E-6;-300.0E-6;SAME;4.1655499640123E+3;INDEPENDENT;AUTO

DISPlay:WAVEView<x>:CURSor:CURSOR<x>? (Query Only)

This query returns the cursor parameters for the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISPlay:WAVEView<x>:CURSor:CURSOR<x>?
Returns	Returns the cursor parameters for the specified cursor in the specified Waveform View.
Examples	DISPlay:WAVEView1:CURSor:CURSOR1? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR:WAVEFORM:BPOSITION 5.0E-6;APOSITION -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:BSOURCE AUTO;HBARS:BPOSITION 9.91E+37;BUNITS "V";AUNITS "V";APOSITION 9.91E+37;DELTA 9.91E+37;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:VBARS: BPOSITION 5.0E-6;APOSITION -5.0E-6;UNITS "s";DELTA 10.0E-6;ALTERNATEB "???? ????";ALTERNATEA "???? ????";:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:STATE 1;FUNCTION WAVEFORM;SCREEN:AYPOSITION -2.0E-3;BXPOSITION -5.0E-6;BYPOSITION -2.0E-3;AXPOSITIO N -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:MODE INDEPENDENT;ASOURCE AUTO;DDT 9.91E+37;ONEOVERDELTAVALUE 100.0E+3;SPLITMODE SAME;LINESTYLE SOLID.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:ASOURCE

This command sets or queries the cursor A source of the specified cursor in the specified Waveform View. `WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.

Group	Cursor
Syntax	<pre>DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:ASOURCE {AUTO CH<x> BUS<x> DCH<x>_DALL MATH<x> REF<x> PLOT<x>} DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:ASOURCE?</pre>
Arguments	<p><code>AUTO</code> specifies auto as the cursor A source.</p> <p><code>CH<x></code> specifies an analog channel to use as the cursor A source.</p> <p><code>BUS<x></code> specifies a bus to use as the cursor A source.</p> <p><code>DCH<x>_DALL</code> specifies a digital channel to use as the cursor A source. The supported digital channel value is 1.</p> <p><code>MATH<x></code> specifies a math waveform to use as the cursor A source.</p> <p><code>REF<x></code> specifies a reference waveform to use as the cursor A source.</p> <p><code>PLOT<x></code> specifies a plot as the cursor A source.</p>
Examples	<pre>DISPlay:WAVEView1:CURSOR:CURSOR1:ASOURCE CH1</pre> <p>sets the cursor 1 source to channel 1.</p> <pre>DISPlay:WAVEView1:CURSOR:CURSOR1:ASOURCE?</pre> <p>might return <code>:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ASOURCE REF1</code> indicating the cursor A source is reference 1.</p>

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:BSOURCE

This command sets or queries the cursor B source of the specified cursor in the specified Waveform View. `WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.

Group	Cursor
Syntax	<pre>DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:BSOURCE {AUTO CH<x> BUS<x> DCH<x>_DALL MATH<x> REF<x> PLOT<x>} DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:BSOURCE?</pre>

Arguments	<p>AUTO specifies auto as the cursor B source.</p> <p>CH<x> specifies an analog channel to use as the cursor B source.</p> <p>BUS<x> specifies a bus to use as the cursor B source.</p> <p>DCH<x>_DALL specifies a digital channel to use as the cursor B source The supported digital channel value is 1.</p> <p>MATH<x> specifies a math waveform to use as the cursor B source.</p> <p>REF<x> specifies a reference waveform to use as the cursor B source.</p> <p>PLOT<x> specifies a plot as the cursor B source.</p>
Examples	<p>DISplay:WAVEview1:CURSor:CURSOR1:BSOURCE CH1 sets the cursor B source to channel 1.</p> <p>DISplay:WAVEview1:CURSor:CURSOR1:BSOURCE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:BSOURCE REF1 indicating the cursor b source is reference 1.</p>

DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT? (Query Only)

This query returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *WAVEview<x>* is the specified Waveform View and must be *WAVEview1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:DDT?
Returns	The delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:DDT? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:DDT 9.91E+37 indicating the delta V over delta T cursor readout value of the specified cursor is 9.91E+37.

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:FUNCTION

This command sets or queries the cursor type of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSOR:CURSOR<x>:FUNCTION {SCREEN WAVEFORM VBARS HBARS} DISplay:WAVEView<x>:CURSOR:CURSOR<x>:FUNCTION?
Arguments	<p>HBARS specifies horizontal bar cursors, which measure in vertical units.</p> <p>VBARS specifies vertical bar cursors, which measure in horizontal units.</p> <p>SCREEN specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.</p> <p>WAVEform specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.</p>
Examples	<p>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION VBARS selects the vertical bar cursor type for the specified Waveform View and cursor.</p> <p>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION HBARS indicating that the specified cursor is set to HBARS.</p>

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:APOSITION

Sets or queries the HBARS vertical A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
--------------	--------

Syntax	<code>DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:APOSITION <NR3></code> <code>DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:APOSITION?</code>
Arguments	<NR3> is the vertical cursor A position of the specified cursor in the specified Waveform View. 0.0 divisions is center, 5.0 top of the waveview, and -5.0 the bottom of the waveview.
Examples	<code>DISPlay:WAVEView1:CURSOR:CURSOR1:HBARS:APOSITION 3.0e0</code> sets the cursor position to 3 V. <code>DISPlay:WAVEView1:CURSOR:CURSOR1:HBARS:APOSITION?</code> might return <code>:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:APOSITION 2.9303448275862</code> indicating the cursor position is 2.93 V.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:AUNITS? (Query Only)

This command queries the cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	<code>DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:AUNITS?</code>
Arguments	<QString> is the cursor A vertical units of the specified cursor in the specified Waveform View.
Examples	<code>DISPlay:WAVEView1:CURSOR:CURSOR1:HBARS:AUNITS?</code> might return <code>:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:AUNITS "v"</code> indicating the AUNITS are set to Volts.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:HBARS:BPOSITION

Sets or queries the HBARS vertical B position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:HBARs:BPOsition <NR3> DISplay:WAVEview<x>:CURSor:CURSOR<x>:HBARs:BPOsition?
Arguments	<NR3> is the vertical cursor B position of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:HBARs:BPOsition 4.0e-1 sets the position to 400 mV. DISplay:WAVEview1:CURSor:CURSOR1:HBARs:BPOsition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BPOSITION 387.9148706896567E-3 indicating the B cursor position is 387.9 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBARs:BUNIts? (Query Only)

This command queries the cursor B vertical units of the specified cursor in the specified Waveform View.

NOTE. *WAVEview<x> is the specified Waveform View and must be WAVEview1. CURSor<x> is the specified cursor and must be CURSOR1.*

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:HBARs:BUNIts?
Returns	<QString> is the cursor B vertical units of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:HBARs:BUNIts? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BUNITS "V" indicating the BUNITS are set to Volts.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBARs:DELTA? (Query Only)

This command queries the delta V cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISPlay:WAVEView<x>:CURSor:CURSOR<x>:HBARs:DELTA?
Returns	The delta V cursor readout value of the specified cursor in the specified Waveform View.
Examples	DISPlay:WAVEView1:CURSor:CURSOR1:HBARs:DELTA? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:DELTA 2.612085129310 indicating the delta between the cursors is 2.61 V.

DISPlay:WAVEView<x>:CURSor:CURSOR<x>:MODE

Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISPlay:WAVEView<x>:CURSor:CURSOR<x>:MODE {INDEPENDENT TRACK}
Arguments	TRACK ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1. INDEPENDENT allows independent adjustment of the two cursors.
Examples	DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK specifies that the cursor positions move in unison. DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK, indicating that the two cursors move in unison.

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:ONEOVERDELTATVALUE? (Query Only)

This query returns the one over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSOR:CURSOR<x>:ONEOVERDELTATVALUE?
Returns	The one over delta T cursor readout value of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEView1:CURSOR:CURSOR1:ONEOVERDELTATVALUE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ONEOVERDELTATVALUE 179.3188E+3 indicating the one over delta time value is 179.3 kHz.

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SCREEN:AXPOSITION

Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SCREEN:AXPOSITION <NR3> DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SCREEN:AXPOSITION?
Arguments	<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.
Examples	DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3 sets the x position of screen cursor1 in waveview1 to 15 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. CURSor<x> is the specified cursor and must be CURSOR1.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition <NR3> DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition?
Arguments	<NR3> the vertical cursor A position of the specified cursor in the specified Waveform View.
Examples	<p>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION 25.0E-3 sets the y position of the specified screen cursor of the specified Waveform View to 25 mV.</p> <p>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION -53.0000E-03 indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.</p>

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition

Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. CURSor<x> is the specified cursor and must be CURSOR1.

Group	Cursor
--------------	--------

Syntax	<code>DISPlay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSITION <NR3></code> <code>DISPlay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSITION?</code>
Arguments	<NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.
Examples	<p><code>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3</code> sets the x position of the specified screen cursor in the specified Waveform View to 15 mV.</p> <p><code>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION?</code> might return <code>:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03</code> indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.</p>

DISPlay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSITION

This command sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. CURSor<x> is the specified cursor and must be CURSOR1.

Group	Cursor
Syntax	<code>DISPlay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSITION <NR3></code> <code>DISPlay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSITION?</code>
Arguments	<NR3> the vertical cursor B position of the specified cursor in the specified Waveform View.
Examples	<p><code>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION 25.0E-3</code> sets the y position of the screen cursor of waveview1 to 25 mV.</p> <p><code>DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION?</code> might return <code>:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION -53.0000E-03</code> indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.</p>

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SPLITMODE

This command sets or queries whether both cursors have the same or different sources.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SPLITMODE {SAME SPLIT} DISplay:WAVEView<x>:CURSOR:CURSOR<x>:SPLITMODE?
Arguments	SAME specifies both cursors have the same source. SPLIT specifies the cursors have different sources.
Examples	DISplay:WAVEView1:CURSOR:CURSOR1:SPLITMODE SPLIT specifies the cursors have different sources. DISplay:WAVEView1:CURSOR:CURSOR1:SPLITMODE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SPLITMODE SAME indicating the cursors have the same source.

DISplay:WAVEView<x>:CURSOR:CURSOR<x>:STATE

This command sets or queries the visible state of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEView<x>:CURSOR:CURSOR<x>:STATE {<NR1> OFF ON} DISplay:WAVEView<x>:CURSOR:CURSOR<x>:STATE?
Arguments	<NR1> = 0 disables the specified cursor in the specified Waveform View; any other value turns this feature on. OFF disables the specified cursor in the specified Waveform View.

ON enables the specified cursor in the specified Waveform View.

- Examples** `DISPlay:WAVEView1:CURSOR:CURSOR1:STATE OFF` turns the specified cursor off.
- `DISPlay:WAVEView1:CURSOR:CURSOR1:STATE?` might return `:DISPlay:WAVEVIEW1:CURSOR:CURSOR1:STATE 1` indicating the specified cursor in the specified Waveform View is on.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:VBARS:APOSITION

This command sets or queries the cursor A horizontal position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

- Group** Cursor
- Syntax** `DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:VBARS:APOSITION <NR3>`
`DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:VBARS:APOSITION?`
- Arguments** <NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.
- Examples** `DISPlay:WAVEView1:CURSOR:CURSOR1:VBARS:APOSITION -4.0e-6` set the cursor position to 4.0 μ s.
- `DISPlay:WAVEView1:CURSOR:CURSOR1:VBARS:APOSITION?` might return `:DISPlay:WAVEVIEW1:CURSOR:CURSOR1:VBARS:APOSITION -5.0E-6` indicating the cursor A position is -5.0 μ s.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:VBARS:BPOSITION

This command sets or queries the cursor B horizontal position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *Cursor<x>* is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:VBArS:BPOsition <NR3> DISplay:WAVEview<x>:CURSor:CURSOR<x>:VBArS:BPOsition?
Arguments	<NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:VBArS:BPOsition 4.0e-6 sets the cursor position to 4.0 μ s. DISplay:WAVEview1:CURSor:CURSOR1:VBArS:BPOsition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:BPOSITION 3.2736951319481E-6 indicating the B cursor position is 3.27 μ s.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArS:DELTA? (Query Only)

This query sets or returns the delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *WAVEview<x> is the specified Waveform View and must be WAVEview1. CURSor<x> is the specified cursor and must be CURSOR1.*

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:VBArS:DELTA?
Returns	The delta T cursor readout value of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:VBArS:DELTA? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:DELTA 6.7926388747725E-6 indicating the delta T cursor readout value is 6.79 μ s.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArS:UNIts? (Query Only)

This query returns cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE. *WAVEview<x>* is the specified Waveform View and must be *WAVEview1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISPlay:WAVEview<x>:CURSOR:CURSOR<x>:VBARS:UNITs?
Returns	The cursor A vertical units of the specified cursor in the specified Waveform View.
Examples	DISPlay:WAVEview1:CURSOR:CURSOR1:VBARS:UNITs? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:UNITS "s" indicating the cursor units are seconds.

DISPlay:WAVEView<x>:CURSOR:CURSOR<x>:WAVEform:APOSITION

This command sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEview<x>* is the specified Waveform View and must be *WAVEview1*.
Cursor<x> is the specified cursor and must be *CURSOR1*.

Group	Cursor
Syntax	DISPlay:WAVEview<x>:CURSOR:CURSOR<x>:WAVEform:APOSITION <NR3> DISPlay:WAVEview<x>:CURSOR:CURSOR<x>:WAVEform:APOSITION?
Arguments	<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.
Examples	DISPlay:WAVEview1:CURSOR:CURSOR1:WAVEform:APOSITION 1.5e-6 sets the position to 1.5 μ s. DISPlay:WAVEview1:CURSOR:CURSOR1:WAVEform:APOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:APOSITION -1.2667480236557E-6 indicating the position is 1.27 μ s.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition? (Query Only)

This command queries the vertical waveform value at the cursor A position in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR:WAVEform:AVPOSition?
Arguments	<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR:WAVEform:AVPOSition? might return :DISplay:WAVEview1:CURSor:CURSOR:WAVEform:AVPOSition 1.35, indicating that the vertical waveform value at the Cursor A position is 1.35 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. CURSor<x> is the specified cursor and must be CURSOR1.*

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR<x>:WAVEform:BPOSition <NR3> DISplay:WAVEview<x>:CURSor:CURSOR<x>:WAVEform:BPOSition?
Arguments	<NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR1:WAVEform:BPOSition 8.0e-9 sets the position to 8.0 ns.

DISplay:WAVEview1:CURSor:CURSOR1:WAVEform:BPOsition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:BPOSITION 711.6310718892624E-9 indicating the cursor position is 711.6 ns.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOsition? (Query Only)

This command queries the vertical waveform value at the cursor B position in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEview1.*

Group	Cursor
Syntax	DISplay:WAVEview<x>:CURSor:CURSOR:WAVEform:BVPOsition?
Arguments	<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.
Examples	DISplay:WAVEview1:CURSor:CURSOR:WAVEform:BVPOsition? might return :DISplay:WAVEview1:CURSor:CURSOR:WAVEform:BVPOsition -0.72, indicating that the vertical waveform value at the Cursor B position is -0.72 V.

DISplay:WAVEView<x>:DCH<x>_D<x>:STATE

This command sets or queries the display state of the specified digital channel in the specified waveview. WAVEView<x> is the specified Waveform View and must be WAVEview1.

Group	Display Control
Syntax	DISplay:WAVEview<x>:DCH<x>_D<x>:STATE {<NR1> OFF ON} DISplay:WAVEview<x>:DCH<x>_D<x>:STATE?
Arguments	DCH<x>_D<x> specifies the digital channel. The supported digital channel value is 1. The supported digital bit values are 0 to 15 and ALL (all digital bits). <NR1> = 0 disables the specified channel on the specified Waveform View; any other value turns this feature on.

OFF disables the display the specified channel on the specified Waveform View.

ON enables the specified channel on the specified Waveform View.

Examples `DISPlay:WAVEView1:DCH1_DALL:STATE ON` enables the display of all bits of digital channel on the specified waveview.

`DISPlay:WAVEView1:DCH1_D0:STATE?` might return
`:DISPLAY:WAVEVIEW1:DCH1_D0:STATE 0` indicating specified digital channels in the specified waveview are off.

DISPLAY:WAVEVIEW<x>:DCH<x>_DALL:DIGORDER

This command sets or queries the order of the digital channels. `WAVEView<x>` is the specified Waveform View and must be `WAVEView1`.

Group Display Control

Syntax `DISPLAY:WAVEVIEW<x>:DCH<x>_DALL:DIGORDER <QString>`
`DISPLAY:WAVEVIEW<x>:DCH<x>_DALL:DIGORDER?`

Arguments `DCH<x>` specifies the digital channel. The supported digital channel value is 1.
`<QString>` specifies the ascending or descending order , enclosed in quotes.

Examples `DISPlay:WAVEView1:DCH1_DALL:DIGORDER "fedcba9876543210"` sets the ascending or descending order of the digital channels.

`DISPlay:WAVEView1:DCH1_DALL:DIGORDER?` might return
`:DISPLAY:WAVEVIEW1:DCH1_DALL:DIGORDER "fedcba9876543210"`
 indicating the order of the digital channels.

DISPlay:WAVEView<x>:DCH<x>_DALL:VERTical:POSition

Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window. `WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `DCH<x>` is the specified digital channel and must be 1.

Group Display Control

Syntax	<code>DISPlay:WAVEView<x>:DCH<x>_DALL:VERTical:POSition <NR3></code> <code>DISPlay:WAVEView<x>:DCH<x>_DALL:VERTical:POSition?</code>
Arguments	<NR3> is the vertical position in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.
Examples	<code>DISPlay:WAVEView1:DCH1_DALL:VERTical:POSition 1.0e0</code> sets the position to 1 division. <code>DISPlay:WAVEView1:DCH1_DALL:VERTical:POSition?</code> might return <code>:DISPlay:WAVEView1:DCH1_DALL:VERTical:POSition -3.4400</code> indicating the vertical position is -3.44 divisions.

DISPlay:WAVEView<x>:FILTer

This command sets or queries the type of interpolation filter for the display.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	<code>DISPlay:WAVEView<x>:FILTer {SINX LINEar}</code>
Arguments	<code>LINEar</code> specifies linear interpolation, where acquired points are connected with straight lines. <code>SINX</code> specifies $\sin(x)/x$ interpolation, where acquired points are fit to a curve.
Examples	<code>DISPlay:WAVEView1:FILTer SINX</code> specifies sine-curve interpolation, when magnifying waveforms. <code>DISPlay:WAVEView1:FILTer?</code> might return <code>:DISPlay:WAVEView1:FILTer LINEar</code> , indicating that straight-line interpolation is specified for magnifying waveforms.

DISPlay:WAVEView<x>:GRAticule

This command selects or queries the type of graticule that is displayed.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISplay:WAVEview<x>:GRATICule {GRID TIME FULL NONE} DISplay:WAVEview<x>:GRATICule?
Arguments	GRID specifies a frame and grid only. TIME specifies a time graticule only. FULL specifies a frame, a grid and cross hairs. NONE specified no graticule.
Examples	DISplay:WAVEview1:GRATICule TIME specifies a time graticule. DISplay:WAVEview1:GRATICule? might return :DISPLAY:WAVEVIEW1:GRATICULE GRID indicating the graticule is a grid.

DISplay:WAVEView<x>:INTENSITY:GRATICule

This command sets or queries the graticule saturation level.

NOTE. *WAVEview<x>* is the specified Waveform View and must be *WAVEview1*.

Group	Display Control
Syntax	DISplay:WAVEview<x>:INTENSITY:GRATICule <NR2> DISplay:WAVEview<x>:INTENSITY:GRATICule?
Arguments	<NR2> is the graticule saturation level.
Examples	DISplay:WAVEview1:INTENSITY:GRATICule 75 sets the saturation level to 75%. DISplay:WAVEview1:INTENSITY:GRATICule? might return :DISPLAY:WAVEVIEW1:INTENSITY:GRATICULE 66.0000 indicating the saturation level is at 66%.

DISplay:WAVEView<x>:INTENSITY:WAVEform

This command sets or queries the waveform saturation level.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISPlay:WAVEView<x>:INTENSITY:WAVEform <NR2> DISPlay:WAVEView<x>:INTENSITY:WAVEform?
Arguments	<NR2> is the waveform saturation level.
Examples	DISPlay:WAVEView1:INTENSITY:WAVEform 75 sets the saturation level to 75%. DISPlay:WAVEView1:INTENSITY:WAVEform? might return :DISPLAY:WAVEVIEW1:INTENSITY:WAVEFORM 62.0000 indicating the saturation level is 62%.

DISPlay:WAVEView<x>:MATH:MATH<x>:AUTOScale

This command sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.

Group	Display Control
Syntax	DISPlay:WAVEView<x>:MATH:MATH<x>:AUTOScale {<NR1> OFF ON} DISPlay:WAVEView<x>:MATH:MATH<x>:AUTOScale?
Arguments	<NR1> = 0 disables the autoscaling the math in the specified Waveform View; any other value turns this feature on. OFF disables the autoscaling the math in the specified Waveform View. ON enables the autoscaling the math in the specified Waveform View.
Examples	DISPlay:WAVEView1:MATH:MATH3:AUTOScale 1 enables the autoscaling the math in the specified Waveform View. DISPlay:WAVEView1:MATH:MATH3:AUTOScale? might return :DISPLAY:WAVEVIEW1:MATH:MATH3:AUTOSCALE 0 indicating that Math 3 will not auto-scale.

DISplay:WAVEView<x>:MATH:MATH<x>:STATE

This command sets or queries the state of the specified math waveform in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.

Group	Display Control
Syntax	DISplay:WAVEView<x>:MATH:MATH<x>:STATE {<NR1> OFF ON} DISplay:WAVEView<x>:MATH:MATH<x>:STATE?
Arguments	<NR1> = 0 disables the specified math in the specified Waveform View; any other value turns this feature on. OFF disables the specified math in the specified Waveform View. ON enables the specified math in the specified Waveform View.
Examples	DISplay:WAVEView1:MATH:MATH1:STATE OFF disables the specified math in the specified Waveform View. DISplay:WAVEView1:MATH:MATH2:STATE? might return :DISPLAY:WAVEVIEW1:MATH:MATH2:STATE 1 indicating the math is displayed.

DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified math waveform in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.

Group	Display Control
Syntax	DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition <NR3> DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition?
Arguments	<NR3> is the vertical position in divisions of the specified math waveform.

Examples `DISPlay:WAVEView1:MATH:MATH1:VERTical:POSition 0` sets the position to 0 divisions.

`DISPlay:WAVEView1:MATH:MATH1:VERTical:POSition?` might return `:DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:POSITION -2.2800` indicating the position is at -2.28 divisions.

DISPlay:WAVEView<x>:MATH:MATH<x>:VERTical:SCALE

Sets or queries the vertical scale of the specified math in volts per division within the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax `DISPlay:WAVEView<x>:MATH:MATH<x>:VERTical:SCALE <NR3>`
`DISPlay:WAVEView<x>:MATH:MATH<x>:VERTical:SCALE?`

Arguments <NR3> is the vertical scale of the specified math waveform.

Examples `DISPlay:WAVEView1:MATH:MATH1:VERTical:SCALE 400e-3` sets the scale to 400 mV per division.

`DISPlay:WAVEView1:MATH:MATH1:VERTical:SCALE?` might return `:DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:SCALE 395.0000E-3` indicating the scale is 395 mV.

DISPlay:WAVEView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax `DISPlay:WAVEView<x>:REF:REF<x>:STATE {<NR1>|OFF|ON}`
`DISPlay:WAVEView<x>:REF:REF<x>:STATE?`

Arguments	<p><NR1> = 0 disables the specified reference in the specified Waveform View; any other value turns this feature on.</p> <p>OFF disables the specified reference in the specified Waveform View.</p> <p>ON enables the specified reference in the specified Waveform View.</p>
Examples	<p>DISPlay:WAVEView1:REF:REF1:STATE OFF disables the specified reference in the specified Waveform View.</p> <p>DISPlay:WAVEView1:REF:REF1:STATE? might return :DISPLAY:WAVEVIEW1:REF:REF1:STATE 1 indicating the specified reference in the specified Waveform View is on.</p>

DISPlay:WAVEView<x>:REF:REF<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified reference in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISPlay:WAVEView<x>:REF:REF<x>:VERTical:POSition <NR3> DISPlay:WAVEView<x>:REF:REF<x>:VERTical:POSition?
Arguments	<NR3> is the vertical position in divisions.
Examples	<p>DISPlay:WAVEView1:REF:REF1:VERTical:POSition -2.5e0 sets the position to -2.5 divisions.</p> <p>DISPlay:WAVEView1:REF:REF1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:POSITION -2.6400 indicating the position is set to -2.64 divisions.</p>

DISPlay:WAVEView<x>:REF:REF<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified reference in volts per div within the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group	Display Control
Syntax	DISplay:WAVEview<x>:REF:REF<x>:VERTical:SCALE <NR3> DISplay:WAVEview<x>:REF:REF<x>:VERTical:SCALE?
Arguments	<NR3> is the vertical scale of the specified reference waveform.
Examples	DISplay:WAVEview1:REF:REF1:VERTical:SCALE 400e-3 sets the scale to 400 mV per division. DISplay:WAVEview1:REF:REF1:VERTical:SCALE? might return :DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:SCALE 378.0000E-3 indicating the scale is 378 mV per division.

DISplay:WAVEView<x>:STYLE

This command sets or queries how the waveforms are displayed for analysis mode.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEview1.*

Group	Display Control
Syntax	DISplay:WAVEview<x>:STYLE {VECTors DOTsonly} DISplay:WAVEview<x>:STYLE?
Arguments	DOTs displays individual data points. New points immediately replace old ones. VECTors connects adjacent data points. New points immediately replace old ones.
Examples	DISplay:WAVEview1:STYLE VECTORS sets the display to connect adjacent data points. DISplay:WAVEview1:STYLE? might return :DISplay:WAVEview1:STYLE DOTs, indicating that data points are not connected.

DISplay:WAVEView<x>:VIEWStyle

The command sets or queries the waveform layout style used by the display.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEview1.*

Group	Display Control
Syntax	DISplay:WAVEview<x>:VIEWStyle {OVERlay STACKed} DISplay:WAVEview<x>:VIEWStyle?
Arguments	OVERlay specifies that the display view style used by the specified Waveform View is overlay. STACKed specifies that the display view style used by the specified Waveform View is stacked.
Examples	DISplay:WAVEview1:VIEWStyle OVERLAY sets the view style to overlay. DISplay:WAVEview1:VIEWStyle? might return :DISPLAY:WAVEVIEW1:VIEWSTYLE STACKED indicating the view style is stacked.

DISplay:WAVEView<x>:Zoom? (Query Only)

This query returns the zoom parameters of the specified Waveform View.

NOTE. *WAVEview<x> is the specified Waveform View and must be WAVEview1.*

Group	Zoom
Syntax	DISplay:WAVEview<x>:Zoom?
Returns	Returns the zoom parameters of the specified Waveform View.
Examples	DISplay:WAVEview1:Zoom? might return :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>? (Query Only)

This query returns the zoom parameters of the specified zoom in the specified Waveform View. <x> must be 1.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
ZOOM<x> is the specified zoom and must be *ZOOM1*.

Group	Zoom
Syntax	DISPlay:WAVEView<x>:ZOOM:ZOOM<x>?
Returns	Returns the zoom parameters of the specified zoom in the specified Waveform View.
Examples	DISPlay:WAVEView1:ZOOM:ZOOM1? might return :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.

DISPlay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition

Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays. It is freely movable around the acquisition settings (horizontal span). An acquired waveform or reference could extend off screen. The valid zoom area does not care about the waveform itself, only the user setting for acquisition.

For example, if horizontal scale is set to 1 second, position to 50, then the acquisition area will go from -5 s to +5 s. Zoom window 0 will focus on -5 s and zoom area 100 will focus on +5 s. If the instrument is stopped and the scale changed to 0.5 s, there will be data off the ends of the display. However, 0% zoom will put the user focus on -2.5 s, the lower bound of the acquisition span.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
ZOOM<x> is the specified zoom and must be *ZOOM1*.

Group	Zoom
Syntax	DISPlay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition <NR3> DISPlay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition?

Arguments <NR3> is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 2× or greater.

Examples DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 50 sets the horizontal position of the zoom box of waveview1 to 50 so that it is centered horizontally on the screen.

DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION? might return :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 10.0000, indicating that the horizontal position of the zoom box of waveview 1 is centered over the first major graticule division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALE

This command sets or queries the horizontal zoom factor of the specified zoom in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group Zoom

Syntax DISplay:WAVEview<x>:ZOOM:ZOOM<x>:HORizontal:SCALE <NR3>
DISplay:WAVEview<x>:ZOOM:ZOOM<x>:HORizontal:SCALE?

Arguments <NR3> is the amount of expansion in the horizontal direction in 1-2-4 increments of the specified zoom in the specified Waveform View.

Examples DISplay:WAVEview1:ZOOM:ZOOM1:HORizontal:SCALE 5 sets the horizontal zoom factor of zoom1 in waveview1 to 5x.

DISplay:WAVEview1:ZOOM:ZOOM1:HORizontal:SCALE? might return :DISplay:WAVEview1:ZOOM:ZOOM1:HORizontal:SCALE 10, indicating that the zoom factor of waveview1 is set to 10x.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALE

This command sets or queries the overview window horizontal scale in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
ZOOM<x> is the specified zoom and must be *ZOOM1*.

Group	Zoom
Syntax	DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALE <NR3> DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALE?
Arguments	<NR3> is the horizontal scale of the zoom window.
Examples	DISplay:WAVEView1:ZOOM:ZOOM1HORizontal:WINSCALE? might return :DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:WINSCALE 80.0000E-9 indicating that the horizontal scale of the overview window is 80 ns per division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe

This command sets or queries the zoom display state of the specified zoom in the specified Waveform View. This command is equivalent to pushing the zoom button on the front panel.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*.
ZOOM<x> is the specified zoom and must be *ZOOM1*.

Group	Zoom
Syntax	DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe {ON OFF <NR1>} DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe?
Arguments	ON turns the specified zoom on. OFF turns specified zoom off. <NR1> = 0 disables the specified zoom; any other value enables the specified zoom.
Examples	DISplay:WAVEView1:ZOOM:ZOOM1:STATe ON turns the specified zoom on. DISplay:WAVEView1:ZOOM:ZOOM1:STATe? might return :DISplay:WAVEView1:ZOOM:ZOOM1:STATe 1, indicating that specified zoom is on.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition

This command sets or queries the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. It is measured from the top to bottom of the acquisition window. The top of the zoom window is $-5 * \text{vertical zoom factor}$. The bottom of the zoom window is $+5 * \text{the vertical zoom factor}$. For a zoom of 5x, the position ranges from -25 to 25.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *ZOOM<x>* is the specified zoom and must be *ZOOM1*.

Group Zoom

Syntax DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition?

Arguments NR3 is the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. The top of the zoom window is $-5 * \text{vertical zoom factor}$. The bottom of the zoom window is $+5 * \text{the vertical zoom factor}$. For a vertical zoom of 5x, the position ranges from -25 to 25.

Examples DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 50 sets the vertical position of the specified zoom in the specified Waveform View to 50, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition? might return :DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 23.90000, indicating that the vertical position of the specified zoom in the specified Waveform View is set to 23.9, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE

This command sets or queries the vertical zoom factor of the specified zoom in the specified Waveform View.

NOTE. *WAVEView<x>* is the specified Waveform View and must be *WAVEView1*. *ZOOM<x>* is the specified zoom and must be *ZOOM1*.

Group	Zoom
Syntax	DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE <NR3> DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALE?
Arguments	<NR3> is the amount of vertical expansion or compression. Based on the value that you entered, this command uses the nearest scale factor. Setting the vertical scale to 1 indicates unity (no zoom).
Examples	DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALE 5 sets the vertical scale of zoom1 of waveview1 to 5×. DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALE? might return :DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALE 2.0000, indicating that the vertical scale of zoom1 of waveview1 is 2×.

DISplay:WAVEView<y>:REF:REF<x>:FRAMe

This command sets or returns the selected frame of the specified analog ref. Each ref has a unique selected frame.

Group	Display Control
Syntax	DISplay:WAVEView<y>:REF:REF<x>:FRAMe <NR1> DISplay:WAVEView<y>:REF:REF<x>:FRAMe?
Arguments	<NR1> is the selected frame of the specified analog ref.
Examples	DISplay:WAVEView1:REF:REF2:FRAMe? might return :DISplay:WAVEView1:REF:REF2:FRAMe 1, indicating the selected frame is 1.

DISplay:WAVEView<y>:REF<x>_DALL:FRAMe

This command sets or returns the selected frame of the specified digital ref. Each ref has a unique selected frame.

Group	Display Control
--------------	-----------------

Syntax DISPlay:WAVEView<y>:REF<x>_DALL:FRAME <NR1>
 DISPlay:WAVEView<y>:REF<x>_DALL:FRAME?

Arguments <NR1> is the selected frame of the specified digital ref.

Examples DISPlay:WAVEView<y>:REF<x>_DALL:FRAME? might return
 :DISPLAY:WAVEVIEW1:REF1_DALL:FRAME 1, indicating the selected frame
 is 1.

*ESE

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax *ESE <NR1>
 *ESE?

Related Commands [*CLS](#)
[DESE](#)
[*ESR?](#)
[EVENT?](#)
[EVMsg?](#)
[*SRE](#)
[*STB?](#)

Arguments <NR1> specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

The power-on default for the ESER is 0 if *PSC is 1. If *PSC is 0, the ESER maintains the previous power cycle value through the current power cycle.

NOTE. Setting the *DESER* and the *ESER* to the same values allows only those codes to be entered into the Event Queue and summarized on the *ESB* bit (bit 5) of the Status Byte Register. Use the [DESE](#) command to set the *DESER*.

Examples *ESE 209 sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

*ESE? might return 186, showing that the ESER contains [DESE](#) the binary value 10111010.

*ESR? (Query Only)

This query-only command returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax *ESR?

Related Commands [ALLEv?](#)

[*CLS](#)

[DESE](#)

[*ESE](#)

[EVENT?](#)

[EVMsg?](#)

[*SRE](#)

[*STB?](#)

Examples *ESR? might return *ESR 213, showing that the SESR contains the binary value 11010101.

ETHERnet:DHCPbootp

This command sets the network configuration method to DHCP (that is ON) or static IP address (that is OFF).

Group Ethernet

Syntax ETHERnet:DHCPbootp {ON|OFF}
ETHERnet:DHCPbootp?

Arguments ON enables the instrument to search the network for a DHCP server in order to automatically assign a dynamic IP address to the instrument.

NOTE. *Do not use DHCP searching if your instrument has been assigned a static address on a network. If you set this command to ON, the DHCP search will delete or change your static IP address information.*

OFF disables the instrument to search the network for a DHCP server.

Examples ETHERNET:DHCPBOOTP ON sets the instrument to search for a DHCP server and assign a dynamic IP address to the instrument.

ETHERnet:DNS:IPADDRESS

This command specifies the network Domain Name Server (DNS) IP address.

Group Ethernet

Syntax ETHERnet:DNS:IPADDRESS <QString>
ETHERnet:DNS:IPADDRESS?

Arguments <QString> is a standard IP address value, enclosed in quotes.

Examples ETHERNET:DNS:IPADDRESS "128.196.13.252" sets the DNS IP address that the instrument uses to communicate with the network.

ETHERnet:DOMAINname

This command specifies the network domain name.

Group Ethernet

Syntax ETHERnet:DOMAINname <QString>
ETHERnet:DOMAINname?

Arguments <QString> is the network domain name, enclosed in quotes.

Examples `ETHERNET:DOMAINNAME "Alpha1.Mycorp.com"` sets the domain name that the instrument uses to communicate with the network.

ETHERnet:ENET:ADDRESS? (Query Only)

Returns the Ethernet address (MAC address) value assigned to the instrument. This is assigned at the factory and can not be changed.

Group Ethernet

Syntax `ETHERnet:ENET:ADDRESS?`

Examples `ETHERNET:ENET:ADDRESS?` returns an Ethernet address such as `08:00:11:01:02:03`

ETHERnet:GATEWay:IPADDRESS

This command specifies the network gateway IP address.

Group Ethernet

Syntax `ETHERnet:GATEWay:IPADDRESS <QString>`
`ETHERnet:GATEWay:IPADDRESS?`

Related Commands [ETHERnet:NETWORKCONFig](#)
[ETHERnet:IPADDRESS](#)
[ETHERnet:SUBNETMask](#)

Arguments `<QString>` is a standard IP address value, enclosed in quotes.

Examples `ETHERNET:GATEWAY:IPADDRESS "128.143.16.1"` sets the gateway IP address.

ETHERnet:IPADDRESS

This command sets the IP address assigned to the instrument.

Group	Ethernet
Syntax	ETHERnet:IPADDRESS <QString> ETHERnet:IPADDRESS?
Related Commands	ETHERnet:NETWORKCONFig ETHERnet:SUBNETMask ETHERnet:GATEWay:IPADDRESS
Arguments	<QString> is a standard IP address value, enclosed in quotes.
Examples	ETHERNET:IPADDRESS "123.121.13.214" sets the instrument's IP address.

ETHERnet:LXI:LAN:RESET (No Query Form)

This command resets the LXI local area network.

Group	Ethernet
Syntax	ETHERnet:LXI:LAN:RESET
Examples	ETHERnet:LXI:LAN:RESET resets the LXI local area network.

ETHERnet:LXI:LAN:SERVICENAME

This command sets or queries the service name used for the LXI interface.

Group	Ethernet
Syntax	ETHERnet:LXI:LAN:SERVICENAME <QString> ETHERnet:LXI:LAN:SERVICENAME?
Arguments	<QString> is a quoted string of up to 64 characters that specifies the mDNS service name used for the LXI interface.

Examples `ETHERnet:LXI:LAN:SERVICENAME?` might return "Tektronix Oscilloscope MSO2 053CVV"

ETHERnet:LXI:LAN:STATus? (Query Only)

This query returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.

Group Ethernet

Syntax `ETHERnet:LXI:LAN:STATus?`

Related Commands [ETHERnet:PING](#)
[ETHERnet:PING:STATus?](#)

Returns OK — indicates the network is running and the instrument can “see” the network.
FAULT — indicates the network is not visible, or the network settings are incorrect.
IDENTIFY indicates that a message is being displayed on the instrument’s front panel indicating which scope the user is currently accessing.

Examples `ETHER:LXI:LAN:STAT?` might return FAULT, indicating the network is not visible, or the network settings are incorrect.

ETHERnet:NAME

This command sets or queries the instrument Ethernet hostname assigned to the instrument.

Group Ethernet

Syntax `ETHERnet:NAME <QString>`
`ETHERnet:NAME?`

Arguments `<QString>` is the network name assigned to the instrument, enclosed in quotes.

Examples `ETHERNET:NAME "labcscope1"` sets the instrument's network name.

ETHERnet:NETWORKCONFig

This command specifies the Ethernet network configuration setting.

Group Ethernet

Syntax ETHERnet:NETWORKCONFig {AUTOMATIC|MANUAL}
ETHERnet:NETWORKCONFig?

Related Commands [ETHERnet:NAME](#)
[ETHERnet:IPADDRESS](#)
[ETHERnet:SUBNETMASK](#)
[ETHERnet:GATEWAY:IPADDRESS](#)
[ETHERnet:DHCPBOOTP](#)

Arguments AUTOMATIC specifies that the instrument's IP address, subnet mask and gateway settings will be received from a DHCP server on the local network.

MANUAL specifies that the Ethernet settings will be configured manually, using [ETHERnet:IPADDRESS](#), [ETHERnet:SUBNETMASK](#), and [ETHERnet:GATEWAY:IPADDRESS](#).

Examples ETHERnet:NETWORKCONFig MANUAL specifies to configure the Ethernet settings manually.

ETHERnet:NETWORKCONFig? might return AUTOMATIC, indicating the settings are being configured automatically.

ETHERnet:PING (No Query Form)

Sends a ping packet to the instrument gateway and sets the status accordingly.

Group Ethernet

Syntax ETHERnet:PING EXECute

Examples ETHERNET:PING EXECute causes the instrument to ping the gateway IP address.

ETHERnet:PING:STATUs? (Query Only)

Returns the results of sending the [ETHERnet:PING](#) command to ping the gateway IP address.

Group Ethernet

Syntax ETHERnet:PING:STATUs?

Returns OK is returned if the computer at the gateway IP address answers.

NORESPONSE is returned if the computer at the gateway IP address does not answer.

TRYING is returned if the ping operation is still executing.

NEVER is returned if ETHERnet:PING EXECute has not been previously sent, indicating that no ping response has ever been received in response to an ETHERnet:PING EXECute command since the instrument was powered on.

ETHERnet:SUBNETMask

This command sets or queries the instrument subnet mask value.

Group Ethernet

Syntax ETHERnet:SUBNETMask <QString>
ETHERnet:SUBNETMask?

Related Commands [ETHERnet:NETWORKCONFig](#)
[ETHERnet:IPADDDress](#)
[ETHERnet:GATEWay:IPADDDress](#)

Arguments <QString> is the subnet mask value, enclosed in quotes.

Examples ETHERNET:SUBNETMASK "255.255.255.0" sets the subnet mask value using standard IP address notation format.

EVENT? (Query Only)

This query-only command returns an event code from the Event Queue that provides information about the results of the last [*ESR?](#) read. `EVENT?` also removes the returned value from the Event Queue.

Group	Status and Error
Syntax	<code>EVENT?</code>
Related Commands	ALLEv? *CLS DESE *ESE *ESR? EVMsg? *SRE *STB?
Examples	<code>EVENT?</code> might return <code>:EVENT 110</code> , showing that there was an error in a command header.

EVMsg? (Query Only)

This query-only command removes a single event code from the Event Queue that is associated with the results of the last [*ESR?](#) read and returns the event code with an explanatory message. For more information, see [Event Handling](#).

Group	Status and Error
Syntax	<code>EVMsg?</code>
Related Commands	ALLEv? *CLS DESE *ESE

[*ESR?](#)

[EVENT?](#)

[*SRE](#)

[*STB?](#)

Returns The event code and message in the following format:

```
<Event Code><Comma><QString>[<Event Code><Comma><QString>...]<QString>::= <Message>;[<Command>]
```

where <Command> is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command will be returned as possible without exceeding the 60 character limit of the <Message> and <Command> string combined. The command string is right-justified.

Examples EVMSG? might return :EVMSG 110, "Command header error".

EVQty? (Query Only)

This query-only command returns the number of events that are enabled in the queue. This is useful when using the [ALLEv?](#) query, since it lets you know exactly how many events will be returned.

Group Status and Error

Syntax EVQty?

Related Commands [ALLEv?](#)

[EVENT?](#)

[EVMsg?](#)

Examples EVQTY? might return :EVQTY 3, indicating the number of event codes in the Event Queue.

FACTory (No Query Form)

This command (no query form) resets the instrument to its factory default settings. This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Default Setup from the File menu.

This command Performs the following in addition to what is done for the *RST command:

- Clears any pending OPC operations.
- Resets the following IEEE488.2 registers:
 - *ESE 0 (Event Status Enable Register)
 - *SRE 0 (Service Request Enable Register)
 - DESE 255 (Device Event Status Enable Register)
 - *PSC 1 (Power-on Status Clear Flag)
- Deletes all defined aliases.
- Enables command headers (:HEADer 1).

Group Save and Recall

Syntax FACTory

Related Commands [*PSC](#)
[RECALL:SETUp](#)
[*RST](#)

Arguments None

Examples FACTORY resets the instrument to its factory default settings.

FILESystem? (Query Only)

This query-only command returns the directory listing of the current working directory. This query is the same as the FILESystem:DIR? query.

Group File System

Syntax FILESystem?

Related Commands [FILESystem:COpy](#)
[FILESystem:CWD](#)
[FILESystem:DELEte](#)
[FILESystem:DIR?](#)
[FILESystem:READFile](#)
[FILESystem:REName](#)
[FILESystem:WRITEFile](#)

Arguments None.

Examples FILESYSTEM? might return :FILESYSTEM:DIR
"myFile.txt", "myWaveform.wfm".

FILESystem:COpy (No Query Form)

This command (no query form) copies a named file to a new file. The new file might be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

Group File System

Syntax FILESystem:COpy {<source_file_path>,<destination_file_path>}

Related Commands [FILESystem:CWD](#)
[FILESystem:DELEte](#)

Arguments <source_file_path> is a quoted string that defines the file name and path or directory. If the file path is within the current working directory, you need only specify the file name.

<destination_file_path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples `FILESYSTEM:COPY "E:/setup1.set", "E:/SETUPS/setup1.set"` copies the file named setup1.set, on the E drive to a file named setup1.set in the SETUPS directory on the E drive.

`:FILESystem:CWD "E:/SETUPS"`

`:FILESystem:COpy ".", "I:/Archive/SETUPS"`

`:FILESystem:COpy "../ch1.isf", "I:/SavedWfms/ch1_new.isf"`

FILESystem:CWD

This command sets or queries the current working directory. CWD is short for Current Working Directory. It changes the directory (folder) that the other FILESystem commands operate on.

NOTE. *There are three host ports on the front panel (E:, F:, and G:), and two on the back panel (H: and I:).*

Group File System

Syntax `FILESystem:CWD {<new_working_directory_path>}`
`FILESystem:CWD?`

Arguments `<new_working_directory_path>` is a quoted string that defines the current working; a directory name can be up to 128 characters.

Examples `FILESYSTEM:CWD "E:/waveforms/"` changes the current working directory to a folder named "waveforms" on the USB flash drive installed in the "E:" USB Host port.

`FILESYSTEM:CWD?` might return `:FILESYSTEM:CWD "E:/"` Indicates the current working directory is the root folder of the E: USB flash drive port.

FILESystem:DELEte (No Query Form)

This command (no query form) deletes a named file or directory from a mass storage device. Once removed, the data in that file or directory can no longer be accessed. If the specified file is a directory, it must be empty before it can be deleted.

Group File System

Syntax FILESystem:DELEte <file_path>

Related Commands [FILESystem:COPy](#)
[FILESystem:CWD](#)
[FILESystem:RMDir](#)

Arguments <file_path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples FILESYSTEM:DELETE "NOT_MINE.SET" deletes the file named NOT_MINE.SET from the folder referred to by the FILESYSTEM:CWD[FILESystem:CWD](#) command.

FILESystem:DIR? (Query Only)

This query-only command returns a comma separated list of quoted strings. Each string contains the name of a file or directory in the folder referred to by the FILESYSTEM:CWD[FILESystem:CWD](#) command.

Group File System

Syntax FILESystem:DIR?

Related Commands [FILESystem:CWD](#)
[FILESystem:MKDir](#)

Arguments None

Examples FILESYSTEM:DIR? might return :FILESYSTEM:DIR
 "161012_132039_000.wfm","161012_132039_001.wfm","161220_191452.png",
 "161220_191554.csv","170320_132925_000.wfm","170320_132929_000.set",
 "Ch2-Ch3_000.set","E:","F:","G:","H:","I:","J:","Rgjtest_000.set",
 "RgjSetupRuntTrig4chnl
 s_000.set","RgjSetup_000.set","ScrnShot1jeh2_000.wfm","ScrnShot1jeh_000.wfm",
 "ScrnShot_000.png","ScrnShot_000.wfm","Scrn
 _000.set","Scrn_001.set","Scrn_002.set","Scrn_003.set","Scrn_004.set",
 "Wfm_000.wfm","Wfm_001.wfm","Wfm_002.wfm","Wfm_003
 .wfm","Wfm_004.wfm","Wfm_005.wfm","Wfm_006.wfm","digChans_000.wfm".

FILESystem:HOMEDir? (Query Only)

This query returns the current user's home directory.

Group	File System
Syntax	FILESystem:HOMEDir?
Returns	The current user's home directory as a quoted string.
Examples	FILESYSTEM:HOMEDIR? might return: "C: ".

FILESystem:LDIR? (Query Only)

Returns a comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESYSTEM:CWD[FILESystem:CWD](#) command. This is different than the :DIR query in that it provides a long output format with the file size, type, and modification date/time. Each entry is a semicolon separated list: < file name>;<type>;<size in bytes>;<date>;<time>

Group	File System
Syntax	FILESystem:LDIR?
Returns	A comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESystem:CWD command.
Examples	FILESYSTEM:LDIR? might return "tek0000CH1.isf;FILE;20342;2009-05-21;13:58:24", "TEMP;DIR;4096;2009-09-15;06:20:44".

FILESystem:MKDir (No Query Form)

This command (no query form) creates a new directory.

Group	File System
Syntax	FILESystem:MKDir <directory_path>

Related Commands [FILESystem:CWD](#)
[FILESystem:DIR?](#)

Arguments <directory_path> is a quoted string that specifies the directory to create.

Examples FILESYSTEM:MKDIR "E:\NewDirectory" creates the directory named *NewDirectory* at the root of the E drive.

These two commands create the directory MyNewSubDirectory within the existing directory MyDirectory at the root of the E drive:

```
FILESYSTEM:CWD "E:/MyDirectory"; FILESYSTEM:MKDIR
"MyNewSubDirectory"
```

FILESYSTEM:MOUNT:DRIVE

The command form mounts a network drive specified by the quoted string argument. The quoted string argument is a semicolon separated list of the following fields:

- Drive name - The drive name to be mounted. It is a case insensitive single letter followed by a colon. The drive name must be a letter between 'L:' and 'Z:', inclusive.
- Server identity - The server identity is the DNS name of the server or the IP address of the server.
- Path - The path to be mounted (e.g. /level1/level2/mydirectory).
- User name - The user name for the drive.
- User password - The password for the drive.
- Domain name - The domain/workgroup of the target mount.
- Verbose - The verbose option to capture mount failure messages.

Domain name, user name, user password, and verbose are optional and are only used for mounts requiring SMB/CIFS interworking (MS Windows and MacOS).

The query form returns a 0 or 1 to indicate that the drive name (quoted string) is currently mounted or not. A return of 1 indicates the drive is mounted. A return of 0 indicates the drive is not mounted.

Group File System

Syntax FILESYSTEM:MOUNT:DRIVE <QString>
FILESYSTEM:MOUNT:DRIVE? <QString>

Examples `FILESYSTEM:MOUNT:DRIVE "L:;000.000.0.0;/level1/level2/mydir"`
specifies to mount the L: drive using the drive name, server identity, and path.

`FILESYSTEM:MOUNT:DRIVE? "L:"` might return 0, indicating the L: drive is not mounted.

FILESystem:MOUNT:TEKDrive

This command mounts the TekDrive specified by the quoted string arguments.

Group File System

Syntax `FILESystem:MOUNT:TEKDrive <QString>`
`FILESystem:MOUNT:TEKDrive? <QString>`

Arguments The quoted string must contain the following information:

- Drive name - The TekDrive name to be mounted. It is a case insensitive.
- AutoDisconnectMode:
 - **Power Cycle** - Select to unmount the TekDrive after power cycling the oscilloscope. There is no time restriction when this option is selected.
 - **Never** - Select to mount the TekDrive connection permanently.
 - **Custom** - Select to disconnect the TekDrive after a chosen duration. The default selection is Power Cycle.
- RestrictToCurrentIP - Restrict connection to current network IP only. This may be used for additional network security.
- AutoDisconnectTime - Enter the required time for the Auto Disconnect. The TekDrive gets disconnected automatically from the instrument after the specified time. The duration is in hours. The minimum is 0.25 hours and the maximum is 744 hours.

Examples `FILESystem:MOUNT:TEKDrive "TekDrive1;Custom>true;8"` specifies to mount TekDrive1, auto disconnect it after 8 hours, and restrict the connection to current network IP only.

`FILESYSTEM:MOUNT:DRIVE? "TekDrive1"` might return 1, indicating TekDrive1 is mounted.

FILESystem:READFile (No Query Form)

This command writes the contents of the specified file to the current interface. If the specified file does not exist or is not readable, an appropriate error event is posted.

Group File System

Syntax FILESystem:READFile <QString>

Related Commands [FILESystem:CWD](#)

Arguments <QString> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples FILESYSTEM:READFILE "E:/test_data/tek00016CH1.csv" reads the content of the specified file, if the file exists and is readable, and sends the content of the file to the current interface.

FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file or folder.

Group File System

Syntax FILESystem:REName <old_file_path>,<new_file_path>

Related Commands [FILESystem:CWD](#)

Arguments <old_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

<new_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

Examples FILESYSTEM:RENAME "E:/TEK00000.SET", "E:/MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the E drive.

FILESYSTEM:RENAME
"e:/mySettings/tek00000.set", "e:/setup1.set". This example illustrates how to move a file from one folder to another.

FILESystem:RMDir (No Query Form)

This command (no query form) deletes a named directory. The directory must be empty.

Group File System

Syntax FILESystem:RMDir <directory_path>

Related Commands [FILESystem:CWD](#)

Arguments <directory_path> is a quoted string that defines the folder name and path. If the folder path is within the current working directory, you need only specify the folder name.

Examples FILESYSTEM:RMDIR "E:/OldDirectory" removes the directory named OldDirectory from the root of the E drive.

FILESystem:TEKDrive:CODE? (Query Only)

This command returns short code in string format. This code must be entered (or pasted) at <http://drive.tekcloud.com/activate>. After the code is entered click the **Activate** button to complete the mounting of the TekDrive.

Group File System

Syntax FILESystem:TEKDrive:CODE?

Examples FILESYSTEM:TEKDRIVE:CODE? might return "xYxrr", indicating the code string xYxrr needs to be entered at <http://drive.tekcloud.com/activate> and activated.

FILESystem:TEKDrive:CODE:EXPIRYtime? (Query Only)

This command returns expiry time of short code. It is the absolute time that the expiry command returns. For example, if 2:11 pm is the time the user initiated the TekDrive mounting, then the expiry query returns + 5 minutes (2:16 pm).

Group File System

Syntax FILESystem:TEKDrive:CODE:EXPIRYtime?

Examples FILESYSTEM:TEKDRIVE:CODE:EXPIRYTIME? might return “Wed Nov 11 17:29:44 2020”.

FILESystem:TEKDrive:CODE:STATUS? (Query Only)

This command returns status of short code.

Group File System

Syntax FILESystem:TEKDrive:CODE:STATUS?

Returns **Success** indicates the operation was successful.

AccessDenied indicates it is unable to process the request as permission was denied.

Invalid indicates the operation was invalid.

Unknown indicates it is unable to process the request due to an unknown error.

Pending indicates it is able to generate the code.

NetworkNotFound indicates it is unable to connect to internet.

UserCancelled indicates the operation was cancelled by user.

Examples FILESYSTEM:TEKDRIVE:CODE:STATUS? might return **Pending**, indicating it is able to generate the code.

FILESystem:UNMOUNT:DRive (No Query Form)

This command unmounts the USB drive specified by the quoted string argument.

Group File System

Syntax FILESystem:UNMOUNT:DRIVE <QString>

Related Commands

Arguments <QString> is a quoted string that specifies which USB drive to unmount. String is a case insensitive single letter followed by a colon.

Examples FILESYSTEM:UNMOUNT:DRIVE "G:" specifies to unmount the flash drive installed in the right-most USB slot on the front of the instrument.

FILESystem:UNMOUNT:TEKDrive (No Query Form)

This command unmounts the TekDrive specified by the quoted string argument and the drive name is case insensitive.

Group File System

Syntax FILESystem:UNMOUNT:TEKDrive <QString>

Related Commands

Arguments <QString> specifies the TekDrive to unmount.

Examples FILESystem:UNMOUNT:TEKDrive "TekDrive1" specifies to unmount the TekDrive named "TekDrive1".

FILESystem:WRITEFile (No Query Form)

This command (no query form) writes the specified block data to the specified file on the instruments file system. If the destination file cannot be written, an error event is posted.

Group File System

Syntax FILESystem:WRITEFile <file_path>,<data>

Related Commands [FILESystem:CWD](#)

Arguments <file path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.
 <data> is the specified block data to be written.

FPAnel:PRESS (No Query Form)

This command is used to emulate a button press. When used with knob enumerations, this command pushes the knob. Use the [FPAnel:TURN](#) command to emulate knob turns.

Group Miscellaneous

Syntax FPAnel:PRESS {AUTOset| BUS| CH1<x>| CLEAR| DEFaultsetup| FORCetrig| GPKNOB1| GPKNOB2| HORZPOS| MATH| REF| RUNStOp| SETT050| SINGleseq| TOUCHSCREEN| TRIGMode| USER| VERTPOS| VERTSCALE}

Arguments Arguments are instrument buttons.

Examples FPANEL:PRESS FORCETRIG emulates pressing the Force trigger button.

FPAnel:TURN

This command is used to emulate a knob turn. The optional NR1 specifies the number of clicks where negative values indicate counter clockwise. If not specified, the default of 1 click is used indicating the knob is turned clockwise 1 click.

Group Miscellaneous

Syntax FPAnel:TURN {GPKNOB1| GPKNOB2| HORZPOS| HORZScale| TRIGLevel| VERTPOS| VERTSCALE} [, <NR1>]

Arguments Arguments are knobs that turn obtained during startup.
 <NR1> is the number of clicks to turn the knob.

Examples `FPANEL:TURN TRIGLEVEL ,3` emulates turning the trigger Level knob 3 clicks in the clockwise direction.

HEADer

This command sets or queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

NOTE. *This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers.*

Whether the long or short form of header keywords and enumerations are returned is dependent upon the state of `:VERBose`.

Group Miscellaneous

Syntax `HEADer {<NR1>|OFF|ON}`
`HEADer?`

Related Commands [VERBose](#)

Arguments `<NR1> = 0` sets the Response Header Enable State to false; any other value sets this state to true.

`OFF` sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

`ON` sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

Examples `HEADer OFF` specifies that the instrument omits headers on query responses, so that only the argument is returned.

`HEADer?` might return `:HEADer 1`, indicating that the instrument is including headers on applicable query responses.

With `:VERBose ON` and `:HEADer ON`, the `:ACQuire:MODE?` query might return: `:ACQUIRE:MODE SAMPLE`.

With `:VERBose OFF` and `:HEADer ON`, the `:ACQuire:MODE?` query might return: `:ACQ:MOD SAM`

With `:VERBoSe ON` and `:HEADer OFF`, the `:ACQui re:MODE?` query might return: `SAMPLE`

With `:VERBoSe OFF` and `:HEADer OFF`, the `:ACQui re:MODE?` query might return: `SAM`

HORizontal? (Query Only)

Queries the current horizontal settings.

Group Horizontal

Syntax `HORizontal?`

Examples `HORIZONTAL?` might return `:HORIZONTAL:SAMPLERATE 6.2500E+9;SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC;OVERRIDE 1;VALUE 1250;:HORIZONTAL:MODE:MANUAL:CONFIGURE HORIZONTALSCALE;:HORIZONTAL:MAIN:UNITS S;:HORIZONTAL:DIVISIONS 10;RECORDLENGTH 62500;DELAY:TIME 0.0E+0;MODE 1;:HORIZONTAL:POSITION 50.0000;SCALE 1.0000E-6;ACQDURATION 10.0000E-6.`

HORizontal:ACQDURATION? (Query Only)

This query returns the timebase duration.

Group Horizontal

Syntax `HORizontal:ACQDURATION?`

Returns `<NR3>` returns the duration of the acquisition.

Examples `HORIZONTAL:ACQDURATION?` might return `:HORIZONTAL:ACQDURATION 5.0E-9`, indicating the acquisition duration is 5.0 us.

HORizontal:DELay:MODE

This command sets or queries the horizontal delay mode.

Group	Horizontal
Syntax	<code>HORizontal:DElay:MODE {OFF ON <NR1>}</code> <code>HORizontal:DElay:MODE?</code>
Related Commands	HORizontal:POSition
Arguments	<p>OFF sets the Horizontal Delay Mode to off. This causes the <code>HORizontal:POSition</code> command to operate like the HORIZONTAL POSITION knob on the front panel.</p> <p>ON sets the Horizontal Delay Mode to on. This causes the <code>HORizontal:DElay:TIME</code> command to operate like the HORIZONTAL POSITION knob on the front panel.</p> <p><NR1> = 0 sets the Horizontal Delay Mode to off; any other value sets this mode to on.</p>
Examples	<p><code>HORIZONTAL:DElay:MODE OFF</code> sets the Horizontal Delay Mode to off, allowing the horizontal position command to operate like the HORIZONTAL POSITION knob on the front panel.</p> <p><code>HORIZONTAL:DElay:MODE?</code> might return <code>HORIZONTAL:DElay:MODE OFF</code> indicating that the Horizontal Delay Mode is off and that the horizontal position command operates like the HORIZONTAL POSITION knob on the front panel.</p>

HORizontal:DElay:TIME

This command sets or queries the horizontal delay time that is used when delay mode is on.

Group	Horizontal
Syntax	<code>HORizontal:DElay:TIME <NR3></code> <code>HORizontal:DElay:TIME?</code>
Related Commands	HORizontal:POSition
Arguments	NR3 is the delay in seconds.
Examples	<code>HORizontal:DElay:TIME 0.3</code> sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

HORizontal:DIVisions? (Query Only)

This query-only command returns the number of graticule divisions.

Group Horizontal

Syntax HORizontal:DIVisions?

Examples HORIZONTAL:DIVISIONS? might return :HORIZONTAL:DIVISIONS 10.0000, indicating that the waveform is displayed across ten divisions.

HORizontal:MAIn:INTERPRatio? (Query Only)

This query-only command returns the Horizontal interpolation ratio.

Group Horizontal

Syntax HORizontal:MAIn:INTERPRatio?

Examples HORIZONTAL:MAIN:INTERPRATIO? might return :HORIZONTAL:MAIN:INTERPRATIO 10.0.

HORizontal:MODE

This command set or queries the horizontal operating mode.

Group Horizontal

Syntax HORizontal:MODE {AUTO|MANual}
HORizontal:MODE?

Related Commands [HORizontal:MODE:RECOrdlength](#)
[HORizontal:MODE:SAMPLERate](#)
[HORizontal:MODE:SCALE](#)

Arguments	<p>AUTO selects the automatic horizontal model. Auto mode automatically adjusts the sample rate and record length to provide a high acquisition rate in Fast Acq or signal fidelity in analysis. Record length is read only.</p> <p>MANUAL selects the manual horizontal model. Manual mode lets you change the sample rate, horizontal scale, and record length. These values interact. For example, when you change record length then the horizontal scale also changes.</p>
Examples	<p>HORIZONTAL:MODE AUTO sets the horizontal mode to auto.</p> <p>HORIZONTAL:MODE? might return :HORIZONTAL:MODE MANUAL, indicating that the horizontal mode is manual.</p>

HORizontal:MODE:MANual:CONFIGure

Sets or queries which horizontal control (scale or record length) will primarily change when the sample rate is changed in Manual mode. If the selected control (scale or record length) reaches a limit then the unselected control (record length or scale) may also change.

Group	Horizontal
Syntax	<pre>HORizontal:MODE:MANual:CONFIGure {HORizontal:scale RECORDlength} HORizontal:MODE:MANual:CONFIGure?</pre>
Arguments	<p>HORizontal:scale will change when sample rate is adjusted.</p> <p>RECORDlength will change when sample rate is adjusted.</p>
Examples	<p>HORIZONTAL:MODE:MANUAL:CONFIGURE RECORDlength allows the record length be adjusted when sample rate is changed in Manual mode.</p>

HORizontal:MODE:RECOrdlength

This command sets or queries the record length.

Group	Horizontal
Syntax	<pre>HORizontal:MODE:RECOrdlength <NR1> HORizontal:MODE:RECOrdlength?</pre>

Arguments <NR1> is the record length in samples. Manual mode lets you change the record length, while the record length is read only for Automatic mode.

Examples `HORIZONTAL:MODE:RECORDLENGTH 1000` sets the record length to 1000 samples.

`HORIZONTAL:MODE:RECORDLENGTH?` might return
`:HORIZONTAL:MODE:RECORDLENGTH 1000000`, indicating that the record length is set to 1,000,000 samples

HORizontal:MODE:SAMPLERate

This command sets or queries the sample rate.

Group Horizontal

Syntax `HORizontal:MODE:SAMPLERate <NR1>`
`HORizontal:MODE:SAMPLERate?`

Arguments <NR1> is the sample rate in samples per second.

Examples `HORIZONTAL:MODE:SAMPLERATE 1e6` sets the sample rate to 1 million samples per second.

`HORIZONTAL:MODE:SAMPLERATE?` might return
`:HORIZONTAL:MODE:SAMPLERATE 5.0000E+6`, indicating that the sample rate is set to 5 million samples per second.

HORizontal:MODE:SCALE

This command sets or queries the horizontal scale.

Group Horizontal

Syntax `HORizontal:MODE:SCALE <NR1>`
`HORizontal:MODE:SCALE?`

Arguments <NR1> is the horizontal scale in seconds per division.

Examples `HORIZONTAL:MODE:SCALE 2e-9` sets the horizontal scale to 2 ns per division.
`HORIZONTAL:MODE:SCALE?` might return `:HORIZONTAL:MODE:SCALE 20.0000E-6`, indicating that the horizontal scale is set to 10 μ s per division.

HORizontal:POSition

This command sets or queries the horizontal position as a percent of screen width. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel. When Horizontal Delay Mode is turned on, the horizontal position is forced to 50%.

Group Horizontal

Syntax `HORizontal:POSition <NR3>`
`HORizontal:POSition?`

Arguments `<NR3>` is from 0 to \approx 100 and is the position of the trigger point on the screen (0 = left edge, 100 = right edge).

Examples `HORIZONTAL:POSITION 10` sets the trigger position of the waveform such that 10% of the display is to the left of the trigger position.
`HORIZONTAL:POSITION?` might return `:HORIZONTAL:POSITION 5.0000E+01` indicates the trigger point displayed on the screen.

HORizontal:PREViewstate? (Query Only)

This query returns the display system preview state.

Group Horizontal

Syntax `HORizontal:PREViewstate?`

Returns `<NR1>` = 1 if the system is in the preview state.
`<NR1>` = 0 if the system is not in the preview state.

Examples `HORIZONTAL:PREVIEWSTATE?` might return `:HORIZONTAL:PREVIEWSTATE 0` indicating the system is not in the preview state.

HORizontal:RECOrdlength

This command sets or queries the horizontal record length. To change the record length the Horizontal Mode must be set to Manual.

Group	Horizontal
Syntax	HORizontal:RECOrdlength <NR1> HORizontal:RECOrdlength?
Arguments	<NR1> is the horizontal record length.
Examples	HORIZONTAL:RECORDLENGTH 1000 sets the record length to 1000 samples. HORIZONTAL:RECORDLENGTH? might return :HORIZONTAL:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples

HORizontal:SAMPLERate

This command sets or queries the horizontal sample rate.

Group	Horizontal
Syntax	HORizontal:SAMPLERate <NR3> HORizontal:SAMPLERate?
Arguments	<NR3> is the horizontal sample rate in samples per second.
Examples	HORIZONTAL:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second. HORIZONTAL:SAMPLERATE? might return :HORIZONTAL:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide

Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.

Group	Horizontal
--------------	------------

Syntax `HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRIDE`
 `{OFF|ON|0|1}`
 `HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRIDE?`

Arguments 0 does not allow override of the horizontal analyze minimum sample rate.
 1 allows override of the horizontal analyze minimum sample rate.
 OFF does not allow override of the horizontal analyze minimum sample rate.
 ON allows override of the horizontal analyze minimum sample rate.

Examples `HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE OFF` does not allow override of the horizontal analyze minimum sample rate.

`HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE?` might return `:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE 1` indicating that override of the horizontal analyze minimum sample rate is allowed.

HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue

Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.

Group Horizontal

Syntax `HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue`
 `{AUTOMATIC|<NR3>}`
 `HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?`

Related Commands

Arguments `AUTOMATIC` allows the instrument to set the minimum value.
 `<NR3>` is the minimum sample rate.

Examples `HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue AUTOMATIC` allows the instrument to set the minimum value.

`HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?` might return `HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC` indicating the instrument automatically sets the minimum value.

HORizontal:SCAle

This command sets or queries the horizontal scale.

Group	Horizontal
Syntax	HORizontal:SCAle <NR3> HORizontal:SCAle?
Arguments	<NR3> is the horizontal scale in time per division.
Returns	The current horizontal scale is returned.
Examples	HORIZONTAL : SCALE 20e-9 sets the horizontal scale to 20 ns/division. HORIZONTAL : SCALE? might return HORIZONTAL : SCALE 20e-9 indicating the horizontal scale is set to 20 ns/division.

ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware similar to that returned by the *IDN? IEEE488.2 common query but does not include the instrument serial number.

Group	Miscellaneous
Syntax	ID?
Related Commands	*IDN?
Examples	ID? might return ID TEK/MSO24,CF:91.1CT,FV:1.42.5, indicating that the instrument model number is set to MSO24, codes and formats is CF:91.1CT, and firmware version is FV:1.42.5.

*IDN? (Query Only)

This query-only command returns the instrument identification code.

Group	Miscellaneous
Syntax	*IDN?
Related Commands	ID?
Examples	*IDN? might return TEKTRONIX,MSO24,C100123,CF:91.1CT FV:1.42.5, indicating the instrument model number, serial number, codes and formats number, and firmware version number.

LIC:UNINSTALL? (Query Only)

Returns the exit license indicated for the user to return to their TekAMS account. Active licenses can be specified by their nomenclature. TransactionIDs can be used to specify an active license or a previously uninstalled license. In either case, the exit-license is returned as block-data.

Group	Miscellaneous
Syntax	LIC:UNINSTALL? <QString>
Arguments	<QString> is the nomenclature of an active license or a TransactionIDs to specify an active license or a previously uninstalled license.
Returns	The exit-license is returned as block-data.
Examples	LIC:UNINSTALL? "2-ULTIMATE" uninstalls the given license and returns the license block data. LIC:UNINSTALL? "569765772" uninstalls the license with the given transaction ID and returns the license block data.

LICense? (Query Only)

This query-only command returns all license parameters.

Group	Miscellaneous
--------------	---------------

Syntax	LICENSE?
Examples	LICENSE? might return :LICENSE:GMT "2022-04-05T16:35:22-07:00";"TES-AAA6BLR2KJN35XVN";"2-ULTIMATE:License Ultimate Bundle for 2 Series MSO, includes Source and Serial options; Node Locked,LIC2-500:License; 2 Series 500MHz bandwidth; Node Locked";2;0,"No events to report - queue empty".

LICENSE:APPID? (Query Only)

This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.

Group	Miscellaneous
Syntax	LICENSE:APPID? {<QString>}
Returns	This query returns a comma-separated list of the active application IDs. If a string argument is provided, a “0” or “1” is returned, according to whether the string matches an active application ID.
Examples	LIC:APPID? might return "BW2-500 , SRAUTO , SRAUTOSEN , SRCOMP , SREMBD", which is a complete list of the active applications. LIC:APPID? "BW2-500" would return “1” because the app is active.

LICENSE:COUNT? (Query Only)

This query returns a count of the number of active licenses installed.

Group	Miscellaneous
Syntax	LICENSE:COUNT?
Returns	A count of the number of active licenses installed.

Examples `LICENSE:COUNT?` might return `:LICENSE:COUNT 2` indicating that 2 active licenses are installed

LICense:ERRor? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. This command is an alias for `ALLEV?`.

Group Miscellaneous

Syntax `LICense:ERRor?`

Related Commands [ALLEV?](#)
[*ESR?](#)
[EVMsg?](#)

Examples `LICense:ERRor?` might return 0, "No events to report- queue empty".

LICense:GMT? (Query Only)

This query returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Group Miscellaneous

Syntax `LICense:GMT?`

Returns The GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Examples `LICENSE:GMT?` might return `:LICENSE:GMT`
`"2016-05-23T17:05:10-07:00"` indicating the local GMT time.

LICense:HID? (Query Only)

This query returns the instrument HostID unique identifier.

Group	Miscellaneous
Syntax	LICense:HID?
Returns	The instrument HostID unique identifier.
Examples	LICENSE:HID? might return :LICENSE:HID "TES-AAA6BLR2KJN35XVN"

LICense:INSTall (No Query Form)

This command accepts a <block_data> license and installs it on the instrument. Restarting the instrument may be necessary to fully activate the additional capabilities.

Group	Miscellaneous
Syntax	LICense:INSTall <block_data>
Arguments	<block_data> is the license in block data format.
Examples	LICENSE:INSTALL <block_data>

LICense:ITEM? (Query Only)

This query returns the details pertaining to a specific license. The NR1 argument is zero-indexed. If no argument is provided, zero is assumed.

Group	Miscellaneous
Syntax	LICense:ITEM? <NR1>
Arguments	<NR1> is the zero-indexed argument specifying a specific license.
Returns	The details pertaining to a specific license.

Examples LICENSE:ITEM? 0 might return
"2-ULTIMATE,Fixed,2023-01-20T02:07:18-08:00,846713276,
""AFG,SRAUTO,SRAUTOSEN,SRCOMP,SREMBD"" ,License; Ultimate
Bundle for 2 Series MSO, includes Source and Serial options;
Node Locked"

LICense:LIST? (Query Only)

This query returns the active license nomenclatures as a comma-separated list of strings. Duplicate nomenclatures, that is, the same license but with different expiration dates, are included.

Group Miscellaneous

Syntax LICense:LIST?

Returns The active license nomenclatures as a comma-separated list of strings.

Examples LICENSE:LIST? might return :LICENSE:LIST "2-ULTIMATE:License;
Ultimate Bundle for 2 Series MSO, includes Source and Serial
options; Node Locked,LIC2-500:License; 2 Series 500MHZ
bandwidth; Node Locked".

LICense:VALidate? (Query Only)

This query accepts a license nomenclature as an argument and returns True (1) if that nomenclature is active and any required hardware is installed, or False (0) if either the nomenclature is not active or required hardware is not installed.

Group Miscellaneous

Syntax LICense:VALidate? <QString>

Arguments <QString> is the license nomenclature.

Returns True (1) if that nomenclature is active and any required hardware is installed.
False (0) if either the nomenclature is not active or required hardware is not installed.

Examples LICENSE:VALIDATE? "AFG" might return :LICENSE:VALIDATE "AFG",0 indicating the license is not active.

LOCK

This command enables or disables the touch screen and all front panel buttons and knobs. There is no front panel equivalent.

When the front panel is locked, the front panel commands will not work and will not generate error events. You can work around a locked front panel, by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use TRIGGER:A SETLevel. To force a trigger, you could use TRIGGER FORCE.

Group Miscellaneous

Syntax LOCK {ALL|NONE}
LOCK?

Related Commands [UNLock](#)
[TOUCHSCREEN:STATE](#)

Arguments ALL disables all front panel controls and the touch screen.
NONE enables all front panel controls and the touch screen. The UNLOCK ALL command only unlocks the front panel controls.
If the instrument is in the Remote With Lockout State (RWLS), the LOCK NONE command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Examples LOCK ALL locks the front panel controls and touch screen.
LOCK? might return :LOCK NONE, indicating that the front panel controls are enabled.

*LRN? (Query Only)

This query-only command returns the commands that list the instrument settings, allowing you to record or “learn” the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the *LRN? query. This command is identical to the SET? command.

Group	Miscellaneous
Syntax	*LRN?
Related Commands	SET?
Examples	<p>*LRN? might return the following response: *RST;:PARAMBATCHING 0;:HEADER 1;:VERBOSE 1;:ALIAS:STATE 0;:ACQUIRE:STATE 1;:ACQUIRE:NUMENV INFINITE;:FASTACQ:STATE 0;:FASTACQ:PALETTE TEMPERATURE;:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC;:SELECT:CH1 1;:SELECT:CH2 0;:SELECT:CH3 0;:SELECT:CH4 0;:SELECT:CH5 0;:SELECT:CH6 0;:SELECT:CH7 0;:SELECT:CH8 0;:CH1:BANDWIDTH 250.0000E+6;:CH2:BANDWIDTH 500.0000E+6;:CH3:BANDWIDTH 500.0000E+6;:CH4:BANDWIDTH 500.0000E+6;:CH5:BANDWIDTH 500.0000E+6;:CH6:BANDWIDTH 500.0000E+6;:CH7:BANDWIDTH 120.0000E+6;:CH8:BANDWIDTH 500.0000E+6;:CH1:BANDWIDTH:ENHANCED 0;:CH2:BANDWIDTH:ENHANCED 0;:CH3:BANDWIDTH:ENHANCED 0;:CH4:BANDWIDTH:ENHANCED 0;:CH5:BANDWIDTH:ENHANCED 0;:CH6:BANDWIDTH:ENHANCED 0;:CH7:BANDWIDTH:ENHANCED 0;:CH8:BANDWIDTH:ENHANCED 0;:DATA:DESTINATION REF1;:DATA:ENCDG ASCII;:DATA:SOURCE CH1;:DATA:START 1;:DATA:STOP 1000000;:HORIZONTAL:MODE AUTO;:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE 1;:HORIZONTAL:MODE:MANUAL:CONFIGURE HORIZONTALSCALE;:HORIZONTAL:SAMPLERATE 6.2500E+9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:VERTICAL:SCALE 1.0000;:TRIGGER:A:TYPE EDGE;:ACQUIRE:SEQUENCE:NUMSEQUENCE 1;:ACQUIRE:SEQUENCE:MODE NUMACQS;:ACQUIRE:MODE SAMPLE;:ACQUIRE:STOPAFTER RUNSTOP;:ACQUIRE:FASTACQ:PALETTE TEMPERATURE;:ACQUIRE:NUMAVG 16T.</p>

MAINWindow:BADGe:BRINGtovieW (No Query Form)

This command is used to automatically scroll the specified badge to make it visible. The input to the command is the badge title name.

Group	Miscellaneous
Syntax	MAINwindow:BADGe:BRINGtovieW <Qstring>
Arguments	<Qstring> specifies a badge title name to make the badge visible.

Examples `MAINwindow:BADGe:BRINGtoview "Meas 1"` makes sure the Meas1 badge is shown in the RRB.

MAINWindow:FONTSize (No Query Form)

Sets the font size for UI text elements. Font size range is 12 to 19 points.

Group Miscellaneous

Syntax `MAINwindow:FONTSize <NR1>`

Arguments `<NR1>` specifies the font size.

Examples `MAINwindow:FONTSize 18` sets the UI text size to 18 points.

MAINWindow:RRBDisplaystate (No Query Form)

Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).

Group Miscellaneous

Syntax `MAINwindow:RRBDisplaystate {1|ON|0|OFF}`

Arguments 1 or ON enables Autoset to change vertical settings.
0 or OFF disables Autoset from changing vertical settings.

Examples `MAINwindow:RRBDisplaystate 1` expands (displays) the Results readout bar.

MASK:DELEte (No Query Form)

This command deletes all mask segments of the specified mask test.

Group Mask

Syntax `MASK:DELEte "MASK<x>"`

Arguments MASK<x> specifies the mask test. This argument is enclosed in quotes.

Examples MASK:DELETE "mask2" will delete all mask segments of mask test 2.

MASK:MASK<x>:COUNT? (Query Only)

This command returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.

Group Mask

Syntax MASK:MASK<x>:COUNT?

Arguments MASK<x> specifies the mask test.

Returns <NR1> is the total number of mask hits in all segments and the number of mask hits in each individual mask segment.

Examples MASK:MASK1:COUNT? might return :MASK:MASK1:COUNT:HITS 211;SEG1:HITS;:MASK:MASK1:COUNT:SEG2:HITS 199; MASK:MASK1:COUNT:SEG3:HITS 12, indicating the number of mask hits on segment 1, 2, and 3 for mask test 1.

MASK:MASK<x>:COUNT:HITS? (Query Only)

This command returns the total number of mask hits in all mask segments for the specified mask test.

Group Mask

Syntax MASK:MASK<x>:COUNT:HITS?

Arguments MASK<x> specifies the mask test.

Returns <NR1> is the total number of mask hits in all mask segments.

Examples MASK:MASK1:COUNT:HITS? might return :MASK:MASK1:COUNT:HITS 211, indicating 211 mask hits for mask test 1.

MASK:MASK<x>:DEFinedby

This command sets or queries whether the specified mask is defined by segments or tolerances. Segment masks are defined by one or more polygons. Tolerance masks are defined by specified horizontal and vertical tolerances around the mask source.

Group Mask

Syntax MASK:MASK<x>:DEFinedby {SEGments|TOLerances}
MASK:MASK<x>:DEFinedby?

Related Commands [MASK:MASK<x>:TOLerance:HORizontal](#)
[MASK:MASK<x>:TOLerance:VERTical](#)
[MASK:MASK<x>:TOLerance:UPDatenow](#)

Arguments MASK<x> specifies the mask number.
SEGments defines the mask by segments.
TOLerances defines the mask by horizontal and vertical tolerances around mask source.

Examples MASK:MASK1:Definedby SEGments sets mask 1 to be defined by segments.
MASK:MASK1:Definedby? might return :MASK:MASK1:Definedby TOLerances, indicating mask 1 is defined by the horizontal and vertical tolerances.

MASK:MASK<x>:DISplay

This command sets or queries the display state of the mask used for the specified mask test.

Group Mask

Syntax MASK:MASK<x>:DISplay {ON|OFF}
MASK:MASK<x>:DISplay?

Arguments	MASK<x> specifies the mask test. ON sets the display state of the specified mask to on. OFF sets the display state of the specified mask to off.
Examples	MASK:MASK1:DISPLAY OFF disables the display of the mask used by mask test 1. MASK:MASK1:DISPLAY? might return :MASK:MASK1:DISPLAY OFF, indicating the display state for mask test 1 is off.

MASK:MASK<x>:LIST? (Query Only)

This command queries the list of segments in the mask used by the specified mask test.

Group	Mask
Syntax	MASK:MASK<x>:LIST?
Arguments	MASK<x> specifies the mask test.
Returns	A comma separated list of segment names.
Examples	MASK:MASK1:LIST? might return :MASK:MASK1:LIST SEG1,SEG2,SEG3, indicating that segment 1, 2, and 3 are used by mask test 1.

MASK:MASK<x>:SEG<x>COUNT:HITS? (Query Only)

The command returns the total number of mask hits in the specified mask segment of the specified mask test.

Group	Mask
Syntax	MASK:MASK<x>:SEG<x>COUNT:HITS?
Arguments	MASK<x> specifies the mask test. SEG<x> specifies the mask segment.

Returns <NR1> is the total number of mask hits in the specified segment.

Examples MASK:MASK1:COUNT:SEG2:HITS? might return
:MASK:MASK1:COUNT:SEG2:HITS 211, indicating 211 mask 1 hits in segment 2.

MASK:MASK<x>:SEG<x>:POINTS

This command sets or queries the X/Y coordinates of all points in the designated mask segment. Mask vertices are in time/volts (currently limited to 1024 characters). The set form defines new points in the mask, replacing any existing points.

Group Mask

Syntax MASK:MASK<x>:SEG<x>:POINTS
MASK:MASK<x>:SEG<x>:POINTS?

Arguments MASK<x> specifies the mask test.
SEG<x> specifies the mask segment.

Examples MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0" sets the X/Y coordinates of all the points in segment 1 of mask test 1.
MASK:MASK1:SEG1:POINTS? might return :MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0", indicating the X/Y coordinates of all the points in segment 1 of mask test 1.

MASK:MASK<x>:SOURce

This command sets or queries analog source for the specified mask test.

Group Mask

Syntax MASK:MASK<x>:SOURce {CH<x>|REF<x>|MATH<x>|RFVstime}
MASK:MASK<x>:SOURce?

Arguments Arguments are the available analog sources.
MASK<x> specifies the mask test.

Examples MASK:MASK1:SOURCE CH2 sets the analog source for mask test 1 to Channel 2.
MASK:MASK1:SOURCE? might return :MASK:MASK1:SOURCE CH2, indicating the analog source for mask test 1 is Channel 2.

MASK:MASK<x>:TEST:STATE

This command sets or queries the state of the specified mask test.

Group Mask

Syntax MASK:MASK<x>:TEST:STATE {ON|OFF}
MASK:MASK<x>:TEST:STATE?

Arguments MASK<x> specifies the mask test.
ON sets the mask test state to ON. When the state is ON the Pass/Fail status and hit count information are reset and the mask test is started.
OFF sets the mask test state to OFF. When the mask test completes the state is set to OFF.

Examples MASK:MASK1:TEST:STATE ON sets the state of mask test 1 to ON.
MASK:MASK1:TEST:STATE? might return :MASK:MASK1:TEST:STATE OFF, indicating mask test 1 is OFF.

MASK:MASK<x>:TEST:STATUS? (Query Only)

This command returns the status of the specified mask test.

Group Mask

Syntax MASK:MASK<x>:TEST:STATUS?

Arguments MASK<x> specifies the mask test.

Returns Return values are PASSING, FAILING, PASSED, or FAILED. When the total hit count equals or exceeds the minimum threshold the status changes from PASSING to FAILING. When the test completes the status changes to PASSED or FAILED. Default value when test state is first set ON or after CLEAR is PASSING.

Examples MASK:MASK1:TEST:STATUS? might return :MASK:MASK1:TEST:STATUS PASSED, indicating mask test 1 passed.

MASK:MASK<x>:TEST:THReshold

This command sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.

Group Mask

Syntax MASK:MASK<x>:TEST:THReshold <NR1>
MASK:MASK<x>:TEST:THReshold?

Arguments MASK<x> specifies the mask test.
<NR1> specifies the threshold value.

Examples MASK:MASK1:TEST:THRESHOLD 10 sets the number of waveform violations to 10 for mask test 1 to fail.
MASK:MASK1:TEST:THRESHOLD? might return :MASK:MASK1:TEST:THRESHOLD 10, indicating 10 waveform violations on mask test 1.

MASK:MASK<x>:TOLerance:HORizontal

This command sets or queries the mask horizontal tolerance.

Group Mask

Syntax MASK:MASK<x>:TOLerance:HORizontal <NR3>
MASK:MASK<x>:TOLerance:HORizontal?

Related Commands [MASK:MASK<x>:TOLerance:UPDatenow](#)

Arguments MASK<x> specifies the mask number.
<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

- Examples** MASK:MASK1:TOLerance:HORizontal .200 sets the tolerance of mask 1 to 200 mdiv or 20% of a division. The mask is not recalculated until an UPDatenow command is sent.
- MASK:MASK1:TOLerance:HORizontal? might return
:MASK:MASK1:TOLerance:HORizontal .200, indicating the tolerance of mask 1 is 200 mdiv.

MASK:MASK<x>:TOLerance:UPDatenow (No Query Form)

This command causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.

- Group** Mask
- Syntax** MASK:MASK<x>:TOLerance:UPDatenow
- Related Commands** [MASK:MASK<x>:TOLerance:HORizontal](#)
[MASK:MASK<x>:TOLerance:VERTical](#)
- Arguments** MASK<x> specifies the mask number.
- Examples** MASK:MASK1:TOLerance:UPDatenow recalculates the tolerance of mask 1 with the current horizontal and vertical tolerance settings.

MASK:MASK<x>:TOLerance:VERTical

This command sets or queries the mask vertical tolerance.

- Group** Mask
- Syntax** MASK:MASK<x>:TOLerance:VERTical <NR3>
MASK:MASK<x>:TOLerance:VERTical?
- Related Commands** [MASK:MASK<x>:TOLerance:UPDatenow](#)
- Arguments** MASK<x> specifies the mask number.
<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

Examples MASK:MASK1:TOLerance:VERTical .100 sets the tolerance of mask 1 to 100 mdiv or 10% of a division. The mask is not recalculated until an UPDateNow command is sent.

MASK:MASK1:TOLerance:VERTical? might return :MASK:MASK1:TOLerance:VERTical .100, indicating the tolerance of mask 1 is 100 mdiv.

MASK:TEST:WAVEforms

This command sets or queries the number of waveform acquisitions to test during mask testing. The number of waveforms applies to all mask tests.

Group Mask

Syntax MASK:TEST:WAVEforms <NR1>
MASK:TEST:WAVEforms?

Arguments MASK<x> specifies the mask test.
<NR1> specifies the number of waveform acquisitions.

Examples MASK:TEST:WAVEFORMS 1k sets the number of waveform acquisitions to test to 1000.

MASK:TEST:WAVEFORMS? might return :MASK:TEST:WAVEFORMS 1k, indicating the number of waveform acquisitions to test is 1000.

MATH:ADDNew (No Query Form)

This command adds the specified math.

Group Math

Syntax MATH:ADDNew <QString>

Related Commands [MATH:LIST?](#)
[MATH:DELeTe](#)

Arguments <QString> is the quoted string specifying the math waveform to add. The argument is of the form "MATH<NR1>", where <NR1> is ≥ 1 .

Examples MATH:ADDNEW "MATH2" adds MATH2.

MATH:DELeTe (No Query Form)

This command deletes the specified math.

Group Math

Syntax MATH:DELeTe <QString>

Related Commands [MATH:ADDNew](#)
[MATH:LIST?](#)

Arguments <QString> is a quoted string specifying the math waveform to delete. The quoted string is of the form "MATH<NR1>", where <NR1> is ≥ 1 .

Examples MATH:DELETE "MATH1", which deletes the MATH1 waveform.

MATH:LIST? (Query Only)

This query returns a comma separated list of all currently defined math waveforms.

Group Math

Syntax MATH:LIST?

Related Commands [MATH:ADDNew](#)
[MATH:DELeTe](#)

Returns All currently defined math waveforms.

Examples MATH:LIST? Might return :MATH:LIST MATH1,MATH4, indicating MATH1 and MATH4 are the currently defined math waveforms.

MATH:MATH<x>:AVG:MODE

This command sets or queries the math average mode flag. If the flag is set to 1, math averaging is turned on. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:AVG:MODE {<NR1> | OFF | ON}

Related Commands [MATH:MATH<x>:AVG:WEIGHT](#)

Arguments <NR1> = 0 turns off average mode, and any other integer turns on average mode.
OFF turns off average mode.
ON turns on average mode.

Examples MATH:MATH2:AVG:MODE ON sets the average mode on.
MATH:MATH1:AVG:MODE? might return :MATH:MATH1:AVG:MODE 0, indicating average mode is off.

MATH:MATH<x>:AVG:WEIGHT

This command sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:AVG:WEIGHT <NR1>

Related Commands [MATH:MATH<x>:AVG:MODE](#)

Arguments <NR1> is the number of acquisitions at which the averaging algorithm will begin exponential averaging.

Examples MATH:MATH2:AVG:WEIGHT 143 sets the number of acquisitions needed to begin averaging to 20
MATH:MATH3:AVG:WEIGHT? might return :MATH:MATH3:AVG:WEIGHT 20 indicating the number of acquisitions needed to begin averaging is 8.

MATH:MATH<x>:CAN:SUPPortedfields

This command sets or queries the field type for the math for the bus source for CAN bus. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:CAN:SUPPortedfields {DATA} MATH:MATH<x>:CAN:SUPPortedfields?
Arguments	DATA sets the field type to DATA.
Examples	MATH:MATH2:CAN:SUPPortedfields DATA sets the field type to DATA. MATH:MATH2:CAN:SUPPortedfields? might return :MATH:MATH2:CAN:SUPPortedfields DATA, indicating the field type for MATH2 for the bus source for CAN bus is DATA.

MATH:MATH<x>:DEFine

This command allows you to define new waveforms using mathematical expressions. The query form of this command returns the math definition for the specified math waveform. The math waveform is specified by x.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants.

Math expressions can be simple, such as Ch1, which specifies that a waveform should show the signal source of Channel 1 with no mathematical computation. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources (including other math waveforms), functions, and operands. As an example, you can enter the expression Log(Ch1+Ch2), which specifies that the signals from channels 1 and 2 are to be algebraically added, and the base 10 log of the sum is to be shown as the final math waveform.

Group	Math
Syntax	MATH:MATH<x>:DEFine <QString> MATH:MATH<x>:DEFine?
Arguments	<QString> quoted string argument is the mathematical expression that defines the waveform. MATH:MATH<x>:DEFINE? is for use when the MATH:MATH<x>:TYPE is ADVANCED.

Examples `MATH:MATH2:DEFine "CH1+CH2"` adds the Channel 1 and Channel 2, defines the Math2 waveform to be Channel1 and Channel2 added together.

`MATH:MATH1:DEFine?` might return

`:MATH:MATH1:DEFine "CH2*REF2"` as the expression that defines Math 1 waveform.

MATH:MATH<x>:FUNction

This command sets or queries the basic math arithmetic function. The math waveform is specified by x.

NOTE. *This command does not affect the same Math equation in Advanced math (also accessed via the command `MATH:MATH<x>:DEFINE`).*

Group Math

Syntax `MATH:MATH<x>:FUNction {ADD|SUBtract|MULTiply|DIVide}`

Arguments `ADD` sets the basic math function to add.
 `SUBtract` sets the basic math function to subtract.
 `MULTiply` sets the basic math function to multiply.
 `DIVide` sets the basic math function to divide.

Examples `MATH:MATH2:FUNCTION MULTIPLY` sets the basic math function to multiply.
`MATH:MATH1:FUNCTION?` might return `:MATH:MATH1:FUNCTION ADD` indicating the current basic math function is addition.

MATH:MATH<x>:I2C:SUPPortedfields

This command sets or queries the field type for the math for the bus source for I2C bus. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:I2C:SUPPortedfields {DATA}`
`MATH:MATH<x>:I2C:SUPPortedfields?`

- Arguments** DATA sets the field type to DATA.
- Examples** MATH:MATH2:I2C:SUPportedfields DATA sets the field type to DATA.
MATH:MATH2:I2C:SUPportedfields? might return
:MATH:MATH2:I2C:SUPportedfields DATA, indicating the field
type for MATH2 for the bus source for I2C bus is DATA.

MATH:MATH<x>:INTERpolation

This command sets or queries whether sinc interpolation is enabled for math on bus source. The math waveform is specified by x.

- Group** Math
- Syntax** MATH:MATH<x>:INTERpolation {ON|OFF}
- Arguments** ON indicates that the sinc interpolation is used for math waveform.
OFF indicates no interpolation is used for math waveform. Waveform will appear “stair-steppy” in this case.
- Examples** MATH:MATH2:INTERpolation ON enables sinc interpolation.
MATH:MATH1:INTERpolation? might return :MATH:MATH1:INTERpolation
ON indicating that the sinc interpolation is used for math waveform.

MATH:MATH<x>:GATing

This command specifies or returns the gating setting. It only applies to Math FFT plots. The math waveform is specified by x.

- Group** Math
- Syntax** MATH:MATH<x>:GATing {NONE|SCREEN|CURSor}
MATH:MATH<x>:GATing?
- Related Commands** [MATH:MATH<x>:TYPE](#)

Arguments	<p>NONE turns off math gating.</p> <p>SCREEN turns on gating, using the left and right edges of the screen.</p> <p>CURSOR limits math to the portion of the waveform between the vertical bar cursors, even if they are off screen.</p>
Examples	<p>MATH:MATH3:GATING CURSOR sets the spectral math plot to be gated by the cursors.</p> <p>MATH:MATH2:GATING? might return :MATH:MATH:2:GATING SCREEN which indicates the spectral math plot is gated by the screen.</p>

MATH:MATH<x>:LABel:COLor

This command sets or queries color of the specified math's label. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:LABel:COLor <QString>
Arguments	<QString> is the color of the label. To return the color to the default color, send an empty string as in this example: :MATH:MATH1:LABEL:COLOR "".
Examples	<p>MATH:MATH3:LABEL:COLOR "GREEN" sets the Math 3 label color to green.</p> <p>MATH:MATH1:LABEL:COLOR? might return :MATH:MATH1:LABEL:COLOR "BLUE" indicating the color of the Math1 label is blue.</p>

MATH:MATH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified math label. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:LABel:FONT:BOLD {<NR1> OFF ON}
Arguments	<p><NR1> = 0 turns off bold, and any other integer turns on bold.</p> <p>OFF turns off bold.</p>

ON turns on bold.

Examples `MATH:MATH3:FONT:BOLD ON` set the math 3 label to bold.
`MATH:MATH2:FONT:BOLD?` might return `:MATH:MATH2:FONT:BOLD`
`0` indicating the math 2 label is not currently bold.

MATH:MATH<x>:LABEL:FONT:ITALIC

This command sets or queries italic state of the specified math label. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABEL:FONT:ITALIC {<NR1>|OFF|ON}`

Arguments `<NR1> = 0` turns off italic, and any other integer turns on italic.
`OFF` turns off italic.
`ON` turns on italic.

Examples `MATH:MATH4:FONT:ITALIC ON` set the math 4 label to be italic.
`MATH:MATH1:FONT:ITALIC?` might return `:MATH:MATH1:FONT:ITALIC`
`0` indicating the math 1 label is not currently italic.

`<NR1>` is the font size of the label.

MATH:MATH<x>:LABEL:FONT:SIZE

This command sets or queries font size of the specified math label. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABEL:FONT:SIZE <NR1>`

Arguments

Examples `MATH:MATH4:LABEL:FONT:SIZE 32` sets the math 4 label size to 32.

`MATH:MATH2:LABEL:FONT:SIZE?` might return
`:MATH:MATH2:LABEL:FONT:SIZE 14` indicating the math 2 label size is currently 14.

MATH:MATH<x>:LABel:FONT:TYPE

This command sets or queries font type of the specified math label, such as Arial or Times New Roman. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABel:FONT:TYPE <QString>`

Arguments `<QString>` is the name of the font type.

Examples `MATH:MATH2:LABEL:FONT:TYPE "Serif"` sets the math 2 label font type to Serif.

`MATH:MATH3:LABEL:FONT:TYPE?` might return
`:MATH:MATH3:LABEL:FONT:TYPE "Monospace"` indicating the math 3 font type is currently Monospace.

MATH:MATH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified math label. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}`

Arguments `<NR1> = 0` turns off underline, and any other integer turns on underline.

OFF turns off underline.

ON turns on underline.

Examples `MATH:MATH3:FONT:UNDERLINE ON` sets the math 3 label to be underlined.
`MATH:MATH2:FONT:UNDERLINE?` might return
`:MATH:MATH2:FONT:UNDERLINE 0` indicating the math 2 label is currently not underlined.

MATH:MATH<x>:LABel:NAME

This command sets or queries the label string, which is used for annotating the math waveform on the screen. The math waveform to which the label is attached is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABel:NAME <QString>`
`MATH:MATH<x>:LABel:NAME?`

Arguments `<QString>` specifies the label to annotate the math waveform.

Examples `MATH:MATH2:LABel:NAME "PROBE POINT7"` assigns "Probe point7" Math 2 waveform.
`MATH:MATH2:LABel:NAME?` might return `:MATH:MATH2:LABel:NAME "Probe point7"`, indicating that Probe point 7 is the label for the Math 2 waveform.

MATH:MATH<x>:LABel:XPOS

This command sets or queries the X position of the specified math label. Maths are specified by x.

Group Math

Syntax `MATH:MATH<x>:LABel:XPOS <NR1>`
`MATH:MATH<x>:LABel:XPOS?`

Related Commands [MATH:MATH<x>:LABel:YPOS](#)

Arguments `<NR1>` is the location (in pixels) where the label for the selected math waveform is displayed, relative to the left edge of the display.

- Examples** `MATH:MATH2:LABEL:XPOS 5` moves the waveform label for the Math 2 waveform so that it begins 5 pixels to the right of the left edge of the screen.
- `MATH:MATH2:LABEL:XPOS?` might return `:MATH:MATH2:LABEL:XPOS 2.5`, indicating that the waveform label for the Math 2 waveform is currently 2.5 pixels to the right of the left edge of the display.

MATH:MATH<x>:LABel:YPOS

This command sets or queries the y-position of the specified math label. The Math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LABel:YPOS <NR1>`
`MATH:MATH<x>:LABel:YPOS?`

Related Commands [MATH:MATH<x>:LABel:XPOS](#)

Arguments <NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the baseline of the waveform.

- Examples** `MATH:MATH2:LABEL:YPOS -2.5` moves the waveform label for the Math 2 waveform to 2.5 pixels below the baseline of the waveform.
- `MATH:MATH2:LABEL:YPOS?` might return `:MATH:MATH2:LABEL:YPOS 0`, indicating that the waveform label for the Math 2 waveform is currently located at the baseline of the waveform.

MATH:MATH<x>:LIN:SUPPortedfields

This command sets or queries the field type for the math for the bus source for LIN bus. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:LIN:SUPPortedfields {DATA}`
`MATH:MATH<x>:LIN:SUPPortedfields?`

Arguments DATA sets the field type to DATA.

Examples `MATH:MATH2:LIN:SUPPortedfields DATA` sets the field type to `DATA`.
`MATH:MATH2:LIN:SUPPortedfields?` might return
`:MATH:MATH2:LIN:SUPPortedfields DATA`, indicating the field type for `MATH2` for the bus source for `LIN` bus is `DATA`.

MATH:MATH<x>:PARAllel:SUPPortedfields

This command sets or queries the field type for the math for the bus source for `PARAllel` bus. The math waveform is specified by `x`.

Group `Math`

Syntax `MATH:MATH<x>:PARAllel:SUPPortedfields {DATA}`
`MATH:MATH<x>:PARAllel:SUPPortedfields?`

Arguments `DATA` sets the field type to `DATA`.

Examples `MATH:MATH2:PARAllel:SUPPortedfields DATA` sets the field type to `DATA`.
`MATH:MATH2:PARAllel:SUPPortedfields?` might return
`:MATH:MATH2:PARAllel:SUPPortedfields DATA`, indicating the field type for `MATH2` for the bus source for `PARAllel` bus is `DATA`.

MATH:MATH<x>:RS232C:SUPPortedfields

This command sets or queries the field type for the math for the bus source for `RS232C` bus. The math waveform is specified by `x`.

Group `Math`

Syntax `MATH:MATH<x>:RS232C:SUPPortedfields {DATA|TXData|RXData}`
`MATH:MATH<x>:RS232C:SUPPortedfields?`

Arguments `DATA` sets the field type to `DATA`.

`TXData` sets the field type to `TXData`. `TXData` field can be set when `RS232` Bus configuration for `Data Inputs` is set to `Two`.

`RXData` sets the field type to `RXData`. `RXData` field can be set when `RS232` Bus configuration for `Data Inputs` is set to `Two`.

Examples `MATH:MATH2:RS232C:SUPPortedfields DATA` sets the field type to DATA.
`MATH:MATH2:RS232C:SUPPortedfields?` might return
`:MATH:MATH2:RS232C:SUPPortedfields DATA`, indicating the
field type for MATH2 for the bus source for RS232C bus is DATA.

MATH:MATH<x>:SENT:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SENT bus. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:SENT:SUPPortedfields`
`{FCData|FCDFirst|FCDTwo|SDATA}`
`MATH:MATH<x>:SENT:SUPPortedfields?`

Arguments `FCData` sets the field type to FCData. FCData field is available to select when SENT Bus configuration for Fast Data Channels is set to 2.

`FCDFirst` sets the field type to FCDFirst.

`FCDTwo` sets the field type to FCDTwo.

`SDATA` sets the field type to SDATA. SDATA is available when SENT Bus configuration for Slow Channel is not None.

Examples `MATH:MATH2:SENT:SUPPortedfields FCData` sets the field type to FCData.
`MATH:MATH2:SENT:SUPPortedfields?` might return
`:MATH:MATH2:SENT:SUPPortedfields FCData`, indicating the
field type for MATH2 for the bus source for SENT bus is FCData.

MATH:MATH<x>:SIGNeddata

This command sets or queries value to denote that bus field is decoded as signed/unsigned data for math on bus source. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:SIGNeddata {ON|OFF}`
`MATH:MATH<x>:SIGNeddata?`

Arguments ON indicates that the bus field is decoded as signed data for drawing the math waveform.

OFF indicates that the bus field is decoded as unsigned data for drawing the math waveform.

Examples MATH:MATH2:SIGNEDdata ON turns on signed data for MATH2 waveform.

MATH:MATH2:SIGNEDdata? might return :MATH:MATH2:SIGNEDdata ON, indicating that the bus field is decoded as signed data for drawing the math waveform.

MATH:MATH<x>:SOURCE<x>

This command sets or queries the specified math source. The source in the command can be either 1 or 2. This command sets the Basic Math components in the user interface, with two sources and a function. You would also need to set the math type to Basic to see the change in the user interface but this will not effect the programmable interface. The math waveform and source are specified by x.

Group Math

Syntax MATH:MATH<x>:SOURCE<x> {CH<x>|MATH<x>|REF<x>}

Related Commands [MATH:MATH<x>:TYPE](#)
[MATH:MATH<x>:FUNCTION](#)

Arguments Arguments are possible math sources. SOURCE1 and SOURCE2 are for use when the MATH:MATH<x>:TYPE is BASIC.

Examples MATH:MATH2:SOURCE1 CH1 sets the first source of math 2 to Channel 1.

MATH:MATH3:SOURCE2? might return :MATH:MATH3:SOURCE2 REF3 indicating the second source of Math 3 is Ref 3.

MATH:MATH<x>:SPECTral:WINDOW

This command sets or queries the window function used to apply the specified FFT window to the input data for the specified math waveform. The Math waveform is specified by x. A spectral window determines what the filter shape of the spectral analyzer will be in the frequency domain. It can be described by

a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer.

Following is a list of arguments that specify the window function used to multiply the input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth).

Group	Math
Syntax	MATH:MATH<x>:SPECTRAL:WINDOW {RECTANGular HAMMING HANNing BLACKMANHarris KAISERBessel GAUSSian FLATTOP2 TEKEXPonential} MATH:MATH<x>:SPECTRAL:WINDOW?
Related Commands	MATH:MATH<x>:TYPE
Arguments	<p>RECTANGular window function is equivalent to multiplying all gate data by one.</p> <p>HAMMING window function is based on a cosine series.</p> <p>HANNing window function is based on a cosine series.</p> <p>BLACKMANHarris window function is based on a cosine series.</p> <p>KAISERBessel window function is based on a cosine series.</p> <p>GAUSSian window function has the best localization characteristics in the joint time/frequency plane</p> <p>FLATTOP2 window function is a cosine series window with a flattened frequency response lobe.</p> <p>TEKEXPonential window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.</p>
Examples	<p>MATH2:SPECTRAL:WINDOW GAUSSIAN applies a Gaussian window to the spectral analyzer input data.</p> <p>MATH2:SPECTRAL:WINDOW? might return :MATH2:SPECTRAL:WINDOW BLACKMANHarris, indicating that the window function used to multiply the spectral analyzer input data is the BLACKMANHarris function.</p>

MATH:MATH<x>:SPI:SUPPortedfields

This command sets or queries the field type for the math for the bus source for SPI bus. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:SPI:SUPPortedfields {DATA MOSIdata MISodata} MATH:MATH<x>:SPI:SUPPortedfields?
Arguments	<p>DATA sets the field type to DATA.</p> <p>MOSIdata sets the field type to MOSIdata. MOSIdata field is available to select when SPI Bus configuration for Data Inputs is set to Two.</p> <p>MISodata sets the field type to MISodata. MISodata field is available to select when SPI Bus configuration for Data Inputs is set to Two.</p>
Examples	<p>MATH:MATH2:SPI:SUPPortedfields DATA sets the field type to DATA.</p> <p>MATH:MATH2:SPI:SUPPortedfields? might return :MATH:MATH2:SPI:SUPPortedfields DATA, indicating the field type for MATH2 for the bus source for SPI bus is DATA.</p>

MATH:MATH<x>:TYPE

This command sets or queries the math type. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:TYPE {BASiC FFT ADVanced}
Arguments	<p>BASiC set the type to basic math.</p> <p>FFT sets the type to FFT math, which can use any live analog or reference waveform in the time domain. NOTE. You can also use FFT as part of a math expression by declaring the type</p> <p>ADVanced. See examples for the command MATH:MATH<x>:DEFine.</p> <p>ADVanced sets the type to advanced math.</p>
Examples	<p>MATH:MATH2:TYPE BASIC sets the type of Math 2 to basic.</p> <p>MATH:MATH2:TYPE? might return :MATH:MATH2:TYPE FFT indicating the type of Math 4 is currently FFT.</p>

MATH:MATH<x>:VUNIT

This command specifies or returns the math custom vertical units. The math waveform is specified by x.

Group	Math
Syntax	MATH:MATH<x>:VUNIT <QString>
Arguments	<QString> is the custom vertical units.
Examples	<p>MATH:MATH2:VUNIT "Small"</p> <p>MATH:MATH2:VUNIT? might return :MATH:MATH2:VUNIT "Large" indicating the Math 2 vertical unit is set to "Large".</p>

MEASTABLE:ADDNew (No Query Form)

This command adds a new measurement results view table to the scope application. The results view table can be named through the argument sent to the command.

Group	Measurement
Syntax	MEASTABLE:ADDNew <QString>
Arguments	<QString> specifies a new measurement results view table to the scope application.
Examples	MEASTABLE:ADDNew "TABLE1" adds a new table view by name TABLE1.

MEASTABLE:DELETE (No Query Form)

This command removes the requested measurement results view table from the scope application.

Group	Measurement
--------------	-------------

Syntax MEASTABle:DELETE <QString>

Arguments <QString> specifies a measurement results view table to remove.

Examples MEASTABle:DELETE "TABLE1" removes TABLE1 from the view.

MEASUrement? (Query Only)

This query-only command returns all measurement parameters.

Group Measurement

Syntax MEASUrement?

Examples MEASUREMENT? might return :MEASUREMENT:GATING OFF;IMMED:TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHOD HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHOD HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS2:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS2:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS2:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION.

MEASUrement:ADDMEAS (No Query Form)

This command adds a measurement.

Group Measurement, IMDA

Syntax MEASUREMENT:ADDMEAS {ACRMS |AMPLITUDE |AREA |BASE |BURSTWIDTH |DATARATE |DELAY |FALLSLEWRATE |FALLTIME |FREQUENCY |HIGHTIME |HOLD |LOWTIME |MAXIMUM |MEAN |MINIMUM |NDUTY |NOVERSHOOT |NPERIOD |NWIDTHTH |PDUTY |PERIOD |PHASE |PK2PK |POVERSHOOT |PWIDTH |RISESLEWRATE |RISETIME |RMS |SETUP |SKEW |TIMEOUTSIDELEVEL |TIMETOMAX |TIMETOMIN |TOP}

Arguments ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DELAY is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQUENCY is the reciprocal of Period. This measurement is made on each cycle in the record.

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINimum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDuty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NOvershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

$$\text{Negative Overshoot} = (\text{Base} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{Top}) / \text{Amplitude} \times 100\%$$

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TIMETOMAX Time to Max is the amount of time from the trigger point to the maximum data point. This measurement can be made across the entire record or on each cycle in the record.

TIMETOMIN Time to Min is the amount of time from the trigger point to the minimum data point. This measurement can be made across the entire record or on each cycle in the record.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

Examples `MEASUREMENT:ADDMEAS FREQUENCY` adds a frequency measurement.

MEASUREMENT:ADDNew (No Query Form)

This command adds the specified measurement.

Group Measurement

Syntax `MEASUREMENT:ADDNew "QString"`

Arguments "QString" is the measurement to add. The argument is of the form "MEAS<NR1>" where $NR1 \geq 1$.

Examples `MEASUREMENT:ADDNew "MEAS11"` adds measurement 11.

MEASUREMENT:ANNOtate

This command sets or queries the annotation state for measurements.

Group Measurement

Syntax `MEASUREMENT:ANNOtate {OFF|AUTO}`
`MEASUREMENT:ANNOtate?`

Arguments OFF turns off measurement annotations.
AUTO turns on visible measurement annotations.

Examples `MEASUREMENT:ANNOtate OFF` turns off measurement annotations.
`MEASUREMENT:ANNOtate?` might return `:MEASUREMENT:ANNOtation:State OFF`, indicating that no measurement annotations are active.

MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax `MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLHigh <NR3>`
`MEASUREMENT:CH<x>:REFLevels:ABSolute:FALLHigh?`

Arguments <NR3> is the high reference level, and is the zero percent level when `MEASUREMENT:IMMed:REFLevel:MEthod` is set to Absolute.

Examples `MEASUREMENT:CH2:REFLevels:ABSolute:FALLHigh 1.5` sets the high reference level of the falling edge to 1.5 V.
`MEASUREMENT:CH2:REFLevels:ABSolute:FALLHigh?` might return `:MEASUREMENT:CH2:REFLevels:ABSolute:FALLHigh 1.0000` indicating that the high reference level of the falling edge is 1.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLLOW

This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLLOW <NR3> MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLLOW?
Arguments	<NR3> is the high reference level, and is the zero percent level when MEASUREMENT:IMMED:REFLEVEL:METHOD is set to Absolute.
Examples	MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5 sets the low reference level of the falling edge to 1.5 V. MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5000 indicating that the low reference level of the falling edge is 1.5 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID

This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID <NR3> MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:FALLMID?
Arguments	<NR3> is the mid reference level used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.
Examples	MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0 sets the mid reference level of the falling edge to 0.0 V. MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating that the mid reference level of the falling edge is 0.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS

This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS <NR3>
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:HYSTERESIS?

Arguments <NR3> is the hysteresis value used for autoset.

Examples MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 sets the reference hysteresis level to 30 mV.

:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS? might return
:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating that reference hysteresis level is set to 30 mV.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHIGH

This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHIGH <NR3>
MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEHIGH?

Arguments <NR3> is the high reference level of the rising edge when the source ref level method is set to absolute.

Examples MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.5 sets the high reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH? might return
:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating that the high reference level of the rising edge is 1.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISELOW

This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISELOW <NR3> MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISELOW?
Arguments	<NR3> is the low reference level of the rising edge when the source ref level method is set to absolute.
Examples	MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.5 sets the low reference level of the rising edge to 1.5 V. MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.0000 indicating that the low reference level of the rising edge is 1.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEMID

This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEMID <NR3> MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:RISEMID?
Arguments	<NR3> is the mid reference level of the rising edge when the source ref level method is set to absolute.
Examples	MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0 sets the mid reference level of the rising edge to 0.0 V. MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating that the mid reference level of the rising edge is 0.0 V.

MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE

This command sets or queries the reference level type for the source. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE {SAME UNIQUE} MEASUREMENT:CH<x>:REFLEVELS:ABSOLUTE:TYPE?
Arguments	SAME specifies that the absolute reference levels for the specified measurement channel are the same. UNIQUE specifies that the absolute reference levels for the specified measurement channel are not the same.
Examples	MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME set the reference levels to be the same. MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the reference levels for absolute measurements are the same.

MEASUREMENT:CH<x>:REFLEVELS:BASETOP

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:BASETOP {AUTO MINMAX MEANHISTOGRAM MODEHISTOGRAM} MEASUREMENT:CH<x>:REFLEVELS:BASETOP?
Arguments	AUTO automatically chooses a reference level method. MINMAX specifies that reference levels are relative to the measurement MIN and MAX. MEANHISTOGRAM specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

- Examples** MEASUREMENT:CH2:REFLEVELS:BASETOP MINMAX specifies that reference levels are relative to the measurement MIN and MAX.
- MEASUREMENT:CH2:REFLEVELS:BASETOP? might return
:MEASUREMENT:CH2:REFLEVELS:BASETOP AUTO indicating the reference levels are chosen automatically.

MEASUREMENT:CH<x>:REFLEVELS:METHoD

This command sets or queries the method used to calculate reference levels for the measurement. The channel number is specified by x.

- Group** Measurement
- Syntax** MEASUREMENT:CH<x>:REFLEVELS:METHoD {PERCENT|ABSOLUTE}
MEASUREMENT:CH<x>:REFLEVELS:METHoD?
- Arguments** PERCENT specifies percent reference level units.
ABSOLUTE specifies absolute reference level units.
- Examples** MEASUREMENT:CH2:REFLEVELS:METHoD PERCENT sets reference levels to be calculated in percent.
- MEASUREMENT:CH2:REFLEVELS:METHoD? might return
:MEASUREMENT:CH2:REFLEVELS:METHoD PERCENT indicating the reference levels are calculated in percent.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

- Group** Measurement
- Syntax** MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLHigh <NR3>
MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLHigh?

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 sets the high reference level for the falling edge to 90%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH? might return
:MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the high reference level for the falling edge is set to 90%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLLOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLLOW <NR3>
MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLLOW?

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the low reference level when the measurement Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000 sets the reference levels for the falling edge to 10%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW? might return
:MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference levels for the falling edge is set to 10%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax	<pre>MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLMID <NR3> MEASUREMENT:CH<x>:REFLEVELS:PERCENT:FALLMID?</pre>
Arguments	<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.
Examples	<p>MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID 50.0000 sets the MID reference level for the falling edge to 50%.</p> <p>MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the MID reference level for the falling edge is set to 50%.</p>

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:HYS TERESIS

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The channel number is specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:CH<x>:REFLEVELS:PERCENT:HYS TERESIS <NR3> MEASUREMENT:CH<x>:REFLEVELS:PERCENT:HYS TERESIS?</pre>
Arguments	<NR3> is the hysteresis value used for the autoset.
Examples	<p>MEASUREMENT:CH2:REFLEVELS:PERCENT:HYS TERESIS 5.0000 sets the reference level hysteresis to 5.0 mV.</p> <p>MEASUREMENT:CH2:REFLEVELS:PERCENT:HYS TERESIS? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:HYS TERESIS 5.0000 indicating the reference level hysteresis is set to 5.0 mV.</p>

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEHIGH <NR3> MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEHIGH?
Arguments	<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.
Examples	MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 sets the high reference level for the rising edge to 90%. MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the high reference level for the rising edge is set to 90%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISELOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISELOW <NR3> MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISELOW?
Arguments	<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.
Examples	MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000 sets the reference levels for the rising edge to 10%. MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000 indicating the reference levels for the rising edge is set to 10%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge

when the measurement ref level method is set to percent. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEMID <NR3> MEASUREMENT:CH<x>:REFLEVELS:PERCENT:RISEMID?
Arguments	<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.
Examples	MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 sets the MID reference level for the rising edge to 50%. MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the MID reference level for the rising edge is set to 50%.

MEASUREMENT:CH<x>:REFLEVELS:PERCENT:TYPE

This command specifies or queries the reference level percent type for the measurement. The channel number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:CH<x>:REFLEVELS:PERCENT:TYPE {TENNINETY TWENTYEIGHTY CUSTOM} MEASUREMENT:CH<x>:REFLEVELS:PERCENT:TYPE?
Arguments	TENNINETY specifies reference levels at the 10 and 90% levels. TWENTYEIGHTY specifies reference levels at the 20 and 80% levels. CUSTOM specifies custom reference levels.
Examples	MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TENNINETY sets the reference levels to the 10 and 90% levels. MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TWENTYEIGHTY indicating the reference levels are set to the 20 and 80% levels.

MEASUREMENT:DELETE (No Query Form)

The command deletes the specified measurement.

Group Measurement

Syntax MEASUREMENT:DELETE <QString>

Arguments <QString> is the measurement to delete. Argument is of the form "MEAS<NR1>" where <NR1> is ≥ 1 .

Examples MEASUREMENT:DELETE "MEAS4" deletes measurement 4.

MEASUREMENT:DELETEALL (No Query Form)

This command deletes all the active instances of measurements defined in the scope application.

Group Measurement

Syntax MEASUREMENT:DELETEALL

Examples MEASUREMENT:DELETEALL deletes all the active instances of measurements.

MEASUREMENT:EDGE<x>

This command sets or queries the type of the edge for the measurement.

Group Measurement

Syntax MEASUREMENT:EDGE<x> {RISE | FALL | BOTH}

Arguments FALL specifies the falling edge of the waveform.
RISE specifies the rising edge of the waveform.
BOTH specifies both a rising and falling edge of the waveform.

Examples MEASUREMENT:EDGE2
 MEASUREMENT:EDGE2? might return :MEASUREMENT:EDGE2 BOTH indicating

MEASUREMENT:GATING

This command sets or queries the global gating type for the measurement.

Group Measurement

Syntax MEASUREMENT:GATING {NONE|SCREEN|CURSOR|LOGIC|SEARCH|TIME}
 MEASUREMENT:GATING?

Related Commands [MEASUREMENT:GATING:ENDTIME](#)
[MEASUREMENT:GATING:STARTTIME](#)

Arguments NONE turns off measurement gating.
 SCREEN turns on gating, using the left and right edges of the screen.
 CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.
 LOGIC specifies that measurements are taken only on the portion of the waveform where the logic source is in the active state.
 SEARCH specifies that measurements are taken based on search criteria.
 TIME limits measurements to the portion of the waveform between the Start and End gate times

Examples MEASUREMENT:GATING SCREEN turns on measurement gating, using the left and right edges of the screen.
 MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUREMENT:GATING:ACTIVE

This command sets or queries the global gating active level used for logic gating.

Group	Measurement
Syntax	MEASUREMENT:GATING:ACTIVE {HIGH LOW} MEASUREMENT:GATING:ACTIVE?
Arguments	HIGH specifies the gate is HIGH. LOW specifies the gate is LOW.
Examples	MEASUREMENT:GATING:ACTIVE LOW specifies a low gate. MEASUREMENT:GATING:ACTIVE? might return :MEASUREMENT:GATING:ACTIVE HIGH indicating the gate is high.

MEASUREMENT:GATING:ENDtime

Sets or queries the end gate time for all measurements that use Global gating.

Group	Measurement
Syntax	MEASUREMENT:GATING:ENDtime <NR3> MEASUREMENT:GATING:ENDtime?
Related Commands	MEASUREMENT:GATING
Arguments	<NR3> is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.
Examples	MEASUREMENT:GATING:ENDtime 5.31e-2 sets the global end gate time to 53.1 ms. MEASUREMENT:GATING:ENDtime? might return 1.05 indicating that the global end gate time is 1.05 s

MEASUREMENT:GATING:HYSteresis

This command sets or queries the global gating hysteresis value used for logic gating.

Group	Measurement
--------------	-------------

- Syntax** MEASUREMENT:GATING:HYSTERESIS <NR3>
MEASUREMENT:GATING:HYSTERESIS?
- Arguments** <NR3> is the gating hysteresis.
- Examples** MEASUREMENT:GATING:HYSTERESIS 40.0e-3 sets the hysteresis to 40 mV.
MEASUREMENT:GATING:HYSTERESIS? might return
:MEASUREMENT:GATING:HYSTERESIS 30.0000E-3 indicating the hysteresis is 30 mV.

MEASUREMENT:GATING:LOGICSource

This command sets or queries the gating data source used for logic gating.

- Group** Measurement
- Syntax** MEASUREMENT:GATING:LOGICSource {CH<x>|MATH<x>|REF<x>}
MEASUREMENT:GATING:LOGICSource?
- Arguments** Arguments are the logic gating source.
- Examples** MEASUREMENT:GATING:LOGICSOURCE MATH1 sets the logic source to MATH 1.
MEASUREMENT:GATING:LOGICSOURCE? might return
:MEASUREMENT:GATING:LOGICSOURCE CH6 indicating the logic source is channel 6.

MEASUREMENT:GATING:MIDRef

This command sets or queries the global gating mid ref value used for logic gating.

- Group** Measurement
- Syntax** MEASUREMENT:GATING:MIDRef <NR3>
MEASUREMENT:GATING:MIDRef?
- Arguments** <NR3> is the mid ref value for gating.

Examples `MEASUREMENT:GATING:MIDREF 2.0` set the gating mid ref to 2.0 V.
`MEASUREMENT:GATING:MIDREF?` might return
`:MEASUREMENT:GATING:MIDREF 1.5000` indicating the gating mid ref is 1.5 V.

MEASUREMENT:GATING:SEARCHSOURCE

This command sets or queries the global gating search source when the gating type is search.

Group Measurement

Syntax `MEASUREMENT:GATING:SEARCHSOURCE SEARCH1`
`MEASUREMENT:GATING:SEARCHSOURCE?`

Arguments Argument is the search source.

Examples `MEASUREMENT:GATING:SEARCHSOURCE SEARCH1` set the search source to search 1.
`MEASUREMENT:GATING:SEARCHSOURCE?` might return
`:MEASUREMENT:GATING:SEARCHSOURCE UNDEFINED` indicating the search source is not defined.

MEASUREMENT:GATING:STARTTIME

Sets or queries the start gate time for all measurements that use Global gating.

Group Measurement

Syntax `MEASUREMENT:GATING:STARTTIME <NR3>`
`MEASUREMENT:GATING:STARTTIME?`

Related Commands [MEASUREMENT:GATING](#)

Arguments <NR3> is the time gating start gate time in seconds. The valid range is -10000 s to 10000 s.

Examples `MEASUREMENT:GATING:STARTTIME -1.5e-3` sets the global start gate time to -1.5 ms.

MEASUREMENT:GATING:STARTtime? might return 3.0e-9, indicating that the global start gate time is 3 ns.

MEASUREMENT:INTERP

This command sets or queries the interpolation mode used to locate edge crossings.

Group	Measurement
Syntax	MEASUREMENT:INTERP {AUTO SINX LINEAR} MEASUREMENT:INTERP?
Arguments	AUTO automatically selects the interpolation mode. SINX specifies sin(x)/x interpolation, where acquired points are fit to a curve. LINEAR specifies linear interpolation, where acquired points are connected with straight lines.
Examples	MEASUREMENT:INTERP SINX set the interpolation mode to sin(x)/x. MEASUREMENT:INTERP? might return :MEASUREMENT:INTERP AUTO indicating the interpolation mode is auto.

MEASUREMENT:LIST? (Query Only)

This query returns a comma separated list of all currently defined measurements.

Group	Measurement
Syntax	MEASUREMENT:LIST?
Returns	Returns a list of all currently defined measurements. When no measurements are defined, the command returns NONE.
Examples	MEASUREMENT:LIST? might return :MEASUREMENT:LIST MEAS1,MEAS2,MEAS3,MEAS4,MEAS5,MEAS6 indicating 6 measurements are defined.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLHIGH

This command sets or queries the value used as the high reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLHIGH <NR3> MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLHIGH?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the high reference level, and is the zero percent level when MEASUREMENT:IMMED:REFLEVEL:METHOD is set to Absolute.
Examples	MEASUREMENT:MATH1:REFLEVELS:ABSOLUTE:FALLHIGH 1.5 sets the reference level to 1.5 V. MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLHIGH? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLLOW

This command sets or queries the value used as the low reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLLOW <NR3> MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLLOW?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the high reference level, and is the zero percent level when MEASUREMENT:IMMED:REFLEVEL:METHOD is set to Absolute.
Examples	MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW -1.5 sets the reference level to -1.5 V.

MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW? might return
:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW -1.0000
indicating the reference level is -1.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLMID

This command sets or queries the value used as the mid reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLMID <NR3> MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:FALLMID?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.
Examples	MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID 0.0 sets the reference level to 0.0 V. MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:HYS TERESIS

This command sets or queries the value of the hysteresis of the reference level when the measurement ref level method is set to absolute. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:HYS TERESIS <NR3> MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:HYS TERESIS?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the hysteresis value used for the autose.

Examples `MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTERESIS 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTERESIS?` might return `:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis value is 30.0 mV.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEHIGH

This command sets or queries the value used as the high reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEHIGH <NR3>`
`MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEHIGH?`

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the high reference level, and is the zero percent level when `MEASUREMENT:IMMED:REFLEVEL:METHOD` is set to Absolute.

Examples `MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.5` sets the reference level to 1.5 V.

`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH?` might return `:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the reference level is 1.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISELOW

This command sets or queries the value used as the low reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISELOW <NR3>`
`MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISELOW?`

- Arguments** MEAS<x> specifies the measurement number.
- <NR3> is the high reference level, and is the zero percent level when MEASUREMENT:IMMED:REFLEVEL:METHOD is set to Absolute.
- Examples** MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW -1.5 sets the reference level to -1.5 V.
- MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW -1.0000 indicating the reference level is -1.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEMID

This command sets or queries the value used as the mid reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

- Group** Measurement
- Syntax** MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEMID <NR3>
MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:RISEMID?
- Arguments** MEAS<x> specifies the measurement number.
- <NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Absolute.
- Examples** MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID 30.0E-3 sets the reference level to 30 mV.
- MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:TYPE

This command sets or queries the reference level type for the measurement. The math number is specified by x.

- Group** Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:TYPE {SAME|UNIQUE}`
`MEASUREMENT:MATH<x>:REFLEVELS:ABSOLUTE:TYPE?`

Arguments `MEAS<x>` specifies the measurement number.

`SAME` specifies that the absolute levels are set the same.

`UNIQUE` specifies that the absolute levels can be set independently.

Examples `MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE?` might return
`:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE SAME` indicating that the absolute levels are set the same.

MEASUREMENT:MATH<x>:REFLEVELS:BASETOP

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The math number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:BASETOP`
`{AUTO|MINMAX|MEANHISTOGRAM|MODEHISTOGRAM|EYEHISTOGRAM}`
`MEASUREMENT:MATH<x>:REFLEVELS:BASETOP?`

Arguments `MEAS<x>` specifies the measurement number.

`AUTO` automatically chooses a reference level method.

`MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

`MEANHISTOGRAM` specifies that reference levels are relative to the histogram mean BASE and TOP.

`MODEHISTOGRAM` specifies that reference levels are relative to the histogram mode BASE and TOP.

`EYEHISTOGRAM` specifies that reference levels are relative to the eye histogram BASE and TOP.

Examples `MEASUREMENT:MATH1:REFLEVELS:BASETOP MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

MEASUREMENT:MATH1:REFLEVELS:BASETOP? might return
:MEASUREMENT:MATH2:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUREMENT:MATH<x>:REFLEVELS:METHOD

This command sets or queries the method used to calculate reference levels for the measurement. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:METHOD {PERCENT ABSOLUTE} MEASUREMENT:MATH<x>:REFLEVELS:METHOD?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>PERCENT specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:MATH<x>:REFLEVEL:PERCENT commands.</p> <p>ABSOLUTE specifies that the reference levels are set explicitly using the MEASUREMENT:MATH<x>:REFLEVEL:ABSOLUTE commands. This method is useful when precise values are required.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:METHOD ABSOLUTE specifies that the reference levels are set explicitly</p> <p>MEASUREMENT:MATH2:REFLEVELS:METHOD? might return :MEASUREMENT:MATH2:REFLEVELS:METHOD PERCENT indicating reference levels are in percent relative to HIGH and LOW.</p>

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLHIGH <NR3> MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLHIGH?

Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level when the measurement Ref level method is set to Percent.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 95 sets the reference level to 95% of TOP.</p> <p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is set to 90% of TOP.</p>

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLLOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group	Measurement
Syntax	<p>MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLLOW <NR3></p> <p>MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLLOW?</p>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW 5 sets the reference level to 5% of TOP.</p> <p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference level is set to 10% of TOP.</p>

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLMID <NR3> MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:FALLMID?</pre>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID 50 sets the reference level to 50% of TOP.</p> <p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the reference level is set to 50% of TOP.</p>

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:HYSTERESIS

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The math number is specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:HYSTERESIS <NR3> MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:HYSTERESIS?</pre>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the hysteresis value used for the autoset.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS 2.5 sets the hysteresis to 2.5% of MAX</p> <p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is set to 5% of MAX.</p>

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x. The measurement number is specified by x.

Group Measurement

Syntax MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEHIGH <NR3>
MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEHIGH?

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEHIGH 95 sets the reference level to 95% of TOP.

MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEHIGH? might return
:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEHIGH 90.0000
indicating the reference level is set to 90% of TOP.

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISELOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISELOW <NR3>
MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISELOW?

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples `MEASUREMENT:MATH2REFLEVELS:PERCENT:RISELOW 5` sets the reference level to 5% of TOP.

`MEASUREMENT:MATH2REFLEVELS:PERCENT:RISELOW?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISELOW 10.0000` indicating the reference level is set to 10% of TOP.

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEMID <NR3>`
`MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:RISEMID?`

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples `MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID 50` sets the reference level to 50% of TOP.

`MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID 50.0000` indicating the reference level is set to 50% of TOP.

MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:TYPE

This command specifies or queries the reference level percent type for the measurement. The math number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:TYPE`
`{TENnInety|TWENTyeighty|CUSTom}`
`MEASUREMENT:MATH<x>:REFLEVELS:PERCENT:TYPE?`

Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>TENNINETY sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.</p> <p>TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.</p> <p>CUSTOM allows setting other reference level percents.</p>
Examples	<p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.</p> <p>MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE TENNINETY indicating the reference levels are set to 10%, 50% and 90%.</p>

MEASUREMENT:MEAS<x>:BURSTEDGTYPE

This command sets or queries the burst edge type for the measurement. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:BURSTEDGTYPE {RISE FALL} MEASUREMENT:MEAS<x>:BURSTEDGTYPE?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>RISE specifies a burst with a rising edge.</p> <p>FALL specifies a burst with a falling edge.</p>
Examples	<p>MEASUREMENT:MEAS2:BURSTEDGTYPE FALL specifies a burst with a falling edge.</p> <p>MEASUREMENT:MEAS2:BURSTEDGTYPE? might return :MEASUREMENT:MEAS2:BURSTEDGTYPE RISE indicating a burst with a rising edge.</p>

MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:MAXIMUM?
Returns	The maximum cycle-cycle statistic value for the specified measurement for all acquisitions.
Examples	MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MAXIMUM? might return :MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MAXIMUM 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:MEAN?
Returns	The mean cycle-cycle statistic value for the specified measurement for all acquisitions.
Examples	MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MEAN? might return :MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MEAN 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:MINIMUM? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:MINIMUM?
Returns	The minimum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples `MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MINIMUM?` might return
 `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MINIMUM 9.91E+37.`

MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:PK2PK?`

Returns The peak to peak cycle-cycle statistic value for the specified measurement all acquisitions.

Examples `MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:PK2PK?` might return
 `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:PK2PK 9.91E+37.`

MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:POPULATION? (Query Only)

This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:CCRESULTS:ALLACQS:POPULATION?`

Returns The population of all cycle-cycle statistics for the specified measurement accumulated over all acquisitions since statistics were last reset.

Examples `MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:POPULATION?` might return
 `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:POPULATION 0` indicating no statistics for the measurement.

MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:STDDev?
Returns	The standard deviation cycle-cycle statistic value for the specified measurement all acquisitions.
Examples	MEASUREMENT:MEAS2:CCRESULTS:ALLAcqs:STDDev? might return :MEASUREMENT:MEAS2:CCRESULTS:ALLAcqs:STDDev 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTAcq:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTAcq:MAXimum?
Returns	The maximum cycle-cycle statistic value for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS2:CCRESULTS:CURRENTAcq:MAXimum? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTAcq:MAXIMUM 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTAcq:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
--------------	-------------

Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:MEAN?
Returns	The mean cycle-cycle statistic value for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MEAN? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MEAN 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:MINIMUM? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:MINIMUM?
Returns	The minimum cycle-cycle statistic value for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS2:CCRESULTS:CURRENTACQ:MINIMUM? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MINIMUM 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:PK2PK?
Returns	The peak to peak cycle-cycle statistic value for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:PK2PK? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:PK2PK 9.91E+37.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:POPULATION? (Query Only)

This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:POPULATION?
Returns	The population of the cycle-cycle statistics for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:POPULATION? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:POPULATION 0 indicating the population is 0.

MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:STDDEV? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:CCRESULTS:CURRENTACQ:STDDEV?
Returns	The standard deviation cycle-cycle statistic value for the specified measurement for the current acquisition.
Examples	MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:STDDEV? might return :MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:STDDEV 9.91E+37.

MEASUREMENT:MEAS<x>:DELAY:EDGE<x>

This command sets or queries the 'to edge' type when EDGE<x> is EDGE1 and the 'from edge' type when EDGE<x> is EDG2, for the measurement when the measurement type is DELAY. Measurements are specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MEAS<x>:DELAY:EDGE<x> {FALL RISE BOTH SAMEas OPPOSITEas} MEASUREMENT:MEAS<x>:DELAY:EDGE<x>?</pre>
Related Commands	MEASUREMENT:MEAS<x>:TOEdge
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>FALL specifies the falling edge of the waveform.</p> <p>RISE specifies the rising edge of the waveform.</p> <p>BOTH specifies both a rising and falling edge of the waveform.</p> <p>SAMEas specifies that both edges of the waveform are the same.</p> <p>OPPOSITEas specifies that the edges of the waveform are not the same.</p>
Examples	<pre>MEASUREMENT:MEAS3:DELAY:EDGE2 RISE specifies that the rising edge of the waveform be used for measurement 3. MEASUREMENT:MEAS1:DELAY:EDGE2? might return :MEASUREMENT:MEAS1:DELAY:EDGE2 FALL, indicating that the falling edge of the waveform is being used for measurement 1.</pre>

MEASUREMENT:MEAS<x>:DISPLAYSTAT:ENABLE

This command turns on and off display of statistics in measurement badges in the user interface. This command affects only the display of statistics, basic-statistics are computed regardless of the state of this command. Measurements are specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MEAS<x>:DISPLAYSTAT:ENABLE {OFF ON <NR1>} MEASUREMENT:MEAS<x>:DISPLAYSTAT:ENABLE?</pre>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>OFF turns off the display of statistics in measurement badges.</p> <p>ON turns on the display of statistics in measurement badges.</p>

<NR1> = 0 turns off the display of statistics in the measurement badge, any other value turns on the display of statistics.

Examples MEASUREMENT:MEAS<x>:DISPlaystat:ENABle 0 turns off the display of statistics in the measurement badge.

MEASUREMENT:MEAS<x>:DISPlaystat:ENABle? might return
:MEASUREMENT:MEAS1:DISPLAYSTAT:ENABLE 1 indicating that the display of statistics in the measurement badge is on.

MEASUREMENT:MEAS<x>:EDGEINCre

This command sets or queries the edge increment value for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:EDGEINCre <NR3>
MEASUREMENT:MEAS<x>:EDGEINCre?

Arguments MEAS<x> specifies the measurement number.
<NR3> is the measurements edge increment value.

Examples :MEASUREMENT:MEAS2:EDGEINCre? might return
:MEASUREMENT:MEAS2:EDGEINCre 1.0000 indicating the edge increment value is set to 1.0000.

MEASUREMENT:MEAS<x>:EDGES:FROMLevel

This command sets or queries the 'from level' edge for the measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:EDGES:FROMLevel {MID|LOW|HIGH}
MEASUREMENT:MEAS<x>:EDGES:FROMLevel?

Arguments MEAS<x> specifies the measurement number.
MID specifies the MID level.

HIGH specifies the HIGH level.

LOW specifies the LOW level.

Examples `MEASUREMENT:MEAS1:EDGES:FROMLEVEL HIGH` set the from level to the high level
`MEASUREMENT:MEAS1:EDGES:FROMLEVEL?` might return
`:MEASUREMENT:MEAS1:EDGES:FROMLEVEL LOW` indicating the from level is the low level.

MEASUREMENT:MEAS<x>:EDGES:LEVEL

This sets or queries the level type for the 'time outside level' measurement. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:EDGES:LEVEL {HIGH|LOW|BOTH}`
`MEASUREMENT:MEAS<x>:EDGES:LEVEL?`

Arguments `MEAS<x>` specifies the measurement number.
HIGH specifies the HIGH level.
LOW specifies the LOW level.
BOTH specifies both the HIGH and LOW level.

Examples `MEASUREMENT:MEAS1:EDGES:LEVEL LOW` sets the level to the low level.
`MEASUREMENT:MEAS1:EDGES:LEVEL?` might return
`:MEASUREMENT:MEAS1:EDGES:LEVEL HIGH` indicating the level is set to the high level.

MEASUREMENT:MEAS<x>:EDGES:N

The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod. Measurements are specified by x.

Group Measurement

Syntax	MEASUREMENT:MEAS<x>:EDGES:N <NR3> MEASUREMENT:MEAS<x>:EDGES:N?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the maximum number of edges used by the measurement.
Examples	measurement:meas1:edges:n 2 sets the number of edges to 2. measurement:meas1:edges:n? might return :MEASUREMENT:MEAS1:EDGES:N 6.0000.

MEASUREMENT:MEAS<x>:EDGES:SLEWRATEMethod

This command sets or queries the slew rate method for the measurement. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:EDGES:SLEWRATEMethod {NOMINAL DDR} MEASUREMENT:MEAS<x>:EDGES:SLEWRATEMethod?
Arguments	MEAS<x> specifies the measurement number. NOMINAL specifies the nominal slew rate method. DDR specifies the DDR slew rate method.
Examples	measurement:meas1:edges:slewratemethod NOMINAL specifies the nominal slew rate method. measurement:meas1:edges:slewratemethod? might return :MEASUREMENT:MEAS1:EDGES:SLEWRATEMETHOD NOMINAL indicating the slew rate method is set to NOMINAL.

MEASUREMENT:MEAS<x>:EDGES:TOLevel

This command sets or queries the 'to level' edge for the measurement. Measurements are specified by x.

Group	Measurement
--------------	-------------

Syntax `MEASUREMENT:MEAS<x>:EDGES:TOLevel {HIGH|MID|LOW}`
`MEASUREMENT:MEAS<x>:EDGES:TOLevel?`

Arguments `MEAS<x>` specifies the measurement number.
`HIGH` specifies the HIGH level.
`MID` specifies the MID level.
`LOW` specifies the LOW level.

Examples `MEASUREMENT:MEAS1:EDGES:TOLevel HIGH` specifies the High to level.
`MEASUREMENT:MEAS1:EDGES:TOLevel?` might return
`:MEASUREMENT:MEAS1:EDGES:TOLevel LOW` indicating the to level
edge is set to the Low level.

MEASUREMENT:MEAS<x>:EDGE<x>

This command sets or queries the type of the specified edge, rise or fall, for the measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:EDGE<x> {RISE|FALL|BOTH}`

Arguments `MEAS<x>` specifies the measurement number.
`RISE` specifies the rising edge.
`FALL` specifies the falling edge.
`BOTH` specifies either the rising or falling edge.

Examples `MEASUREMENT:MEAS2:EDGE1?` might return `:MEASUREMENT:MEAS2:EDGE1`
`RISE` indicating that edge 1 of measurement 2 is the rising edge.

MEASUREMENT:MEAS<x>:FAILCount? (Query Only)

This command returns the number of measurement failures, if applicable, for the selected measurement. The measurement number is specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:FAILCOUNT?

Related Commands [ACTONEVENT:ENABLE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEMODE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE](#)

Returns A running sum of the number of violations of the user specified measurement limit.

Examples MEASUREMENT:MEAS2:FAILCOUNT? might return
:MEASUREMENT:MEAS2:FAILCOUNT 2 indicating that measurement 2 has
two measurement failures.

MEASUREMENT:MEAS<x>:FROMEDGE

This command sets or queries the from edge type for the measurement.
Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:FROMEDGE {RISE|FALL|BOTH}
MEASUREMENT:MEAS<x>:FROMEDGE?

Arguments MEAS<x> specifies the measurement number.
FALL specifies the falling edge of the waveform.
RISE specifies the rising edge of the waveform.
BOTH specifies both the rising and falling edges of the waveform.

Examples MEASUREMENT:MEAS1:FROMEDGE RISE specifies the rising edge of the
waveform.
MEASUREMENT:MEAS1:FROMEDGE? might return
:MEASUREMENT:MEAS1:FROMEDGE BOTH indicating both the rising and falling
edges of the waveform.

MEASUREMENT:MEAS<x>:FROMEDGESEARCHDIRECT

This command sets or queries the from edge search direction for the measurement. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:FROMEDGESEARCHDIRECT {FORWARD BACKWARD} MEASUREMENT:MEAS<x>:FROMEDGESEARCHDIRECT?
Arguments	MEAS<x> specifies the measurement number. FORWARD specifies a forward search from the edge. BACKWARD specifies a backward search from the edge.
Examples	MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT BACKWARD specifies a backward search from the edge. MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT? might return MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT FORWARD indicating a forward search from the edge.

MEASUREMENT:MEAS<x>:GATING

This command sets or queries the gating type for the measurement. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING {NONE SCREEN CURSOR LOGIC SEARCH TIME} MEASUREMENT:MEAS<x>:GATING?
Related Commands	MEASUREMENT:MEAS<x>:GATING:ENDTIME MEASUREMENT:MEAS<x>:GATING:STARTTIME
Arguments	MEAS<x> is the measurement number for which to return a value. NONE specifies measurements are taken across the entire record. SCREEN turns on gating, using the left and right edges of the screen.

CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

LOGIC specifies that measurements are taken only when the logical state of other waveforms is true.

SEARCH specifies that measurements are taken only where the results of a user specified search are found.

TIME limits measurements to the portion of the waveform between the Start and End gate times.

Examples `MEASUREMENT:MEAS1:GATING CURSOR` limits measurements to the portion of the waveform between the vertical bar cursor.

`MEASUREMENT:MEAS1:GATING?` might return `:MEASUREMENT:MEAS1:GATING NONE` indicating measurements are taken across the entire record.

MEASUREMENT:MEAS<x>:GATING:ACTIVE

This command sets or queries the gating active level when the gating type is logic. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:GATING:ACTIVE {HIGH|LOW}`
`MEASUREMENT:MEAS<x>:GATING:ACTIVE?`

Arguments `MEAS<x>` specifies the measurement number.

`HIGH` takes a measurement when logic gating is High.

`LOW` takes a measurement when logic gating Low.

Examples `MEASUREMENT:MEAS1:GATING:ACTIVE LOW` specifies taking measurements when logic gating is Low.

`MEASUREMENT:MEAS1:GATING:ACTIVE?` might return `:MEASUREMENT:MEAS1:GATING:ACTIVE HIGH` indicating measurements are taken when logic gating is High.

MEASUREMENT:MEAS<x>:GATING:ENDtime

Sets or queries the end gate time for the measurement when using Local gating.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:ENDtime <NR3> MEASUREMENT:MEAS<x>:GATING:ENDtime?
Related Commands	MEASUREMENT:MEAS<x>:GATING
Arguments	MEAS<x> specifies the measurement number. <NR3> is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.
Examples	MEASUREMENT:MEAS4:GATING:ENDtime 5.31e-2 sets the local end gate time for Meas 4 to 53.1 ms. MEASUREMENT:MEAS6:GATING:ENDtime? might return 1.05, indicating that the local end gate time for Meas 6 is 1.05 s.

MEASUREMENT:MEAS<x>:GATING:GLOBAL

This command sets or queries the gating settings global flag. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:GLOBAL {OFF ON 0 1} MEASUREMENT:MEAS<x>:GATING:GLOBAL?
Arguments	MEAS<x> specifies the measurement number. OFF specifies gate settings can be changed independently for each individual measurement. ON applies global gate settings to all the measurements' gate settings. 0 specifies gate settings can be changed independently for each individual measurement. 1 applies global gate settings to all the measurements' gate settings.
Examples	MEASUREMENT:MEAS1:GATING:GLOBAL OFF specifies gate settings can be changed independently for each individual measurement.

MEASUREMENT:MEAS1:GATING:GLOBAL? might return
:MEASUREMENT:MEAS1:GATING:GLOBAL 1 indicating that gating
settings apply to all measurements.

MEASUREMENT:MEAS<x>:GATING:HYS TERESIS

This command sets or queries the gating hysteresis value when the gating type is logic. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:HYS TERESIS <NR3> MEASUREMENT:MEAS<x>:GATING:HYS TERESIS?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the gating hysteresis.
Examples	MEASUREMENT:MEAS1:GATING:HYS TERESIS 25.0E-3 sets the hysteresis to 25 mV. MEASUREMENT:MEAS1:GATING:HYS TERESIS? might return :MEASUREMENT:MEAS1:GATING:HYS TERESIS 30.0000E-3 indicating the hysteresis is set to 30.0 mV.

MEASUREMENT:MEAS<x>:GATING:LOGICSOURCE

This command sets or queries the gating data source when the gating type is logic. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:LOGICSOURCE {CH<x> MATH<x> REF<x>}
Arguments	MEAS<x> specifies the measurement number. Arguments are the sources for logic gating.
Examples	MEASUREMENT:MEAS1:GATING:LOGICSOURCE CH3 sets the gating logic source to channel 3.

MEASUREMENT:MEAS1:GATING:LOGICSource? might return
:MEASUREMENT:MEAS1:GATING:LOGICSOURCE CH2 indicating the gating
logic source is channel 2.

MEASUREMENT:MEAS<x>:GATING:MIDRef

This command sets or queries the gating mid ref value when the gating type is logic. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:MIDRef <NR3> MEASUREMENT:MEAS<x>:GATING:MIDRef?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the mid ref value for gating.
Examples	MEASUREMENT:MEAS1:GATING:MIDRef 1.0E+0 sets the gating midref to 1.0. MEASUREMENT:MEAS1:GATING:MIDRef? might return :MEASUREMENT:MEAS1:GATING:MIDREF 0.0E+0 indicating the midref value is set to 0.0.

MEASUREMENT:MEAS<x>:GATING:SEARCHSource

This command sets or queries the gating search source when the gating type is search. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:GATING:SEARCHSource SEARCH1
Arguments	MEAS<x> specifies the measurement number. SEARCH1 is the gating source for search gating.
Examples	MEASUREMENT:MEAS1:GATING:SEARCHSource SEARCH1 sets the gating search source to SEARCH1.

MEASUREMENT:MEAS1:GATING:SEARCHSOURCE? might return
:MEASUREMENT:MEAS1:GATING:SEARCHSOURCE SEARCH1 indicating the
gating search source is SEARCH1.

MEASUREMENT:MEAS<x>:GATING:STARTtime

Sets or queries the start gate time for the measurement when using Local gating.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:GATING:STARTtime <NR3>
MEASUREMENT:MEAS<x>:GATING:STARTtime?

Related Commands [MEASUREMENT:MEAS<x>:GATING](#)

Arguments MEAS<x> specifies the measurement number.

<NR3> is the time gating start gate time in seconds. The valid range is -10000
s to 10000 s.

Examples MEASUREMENT:MEAS1:GATING:STARTtime -1.5e-3 sets the local start gate
time for Meas 1 to -1.5 ms.

MEASUREMENT:MEAS1:GATING:STARTtime? might return 3.0e-9, indicating
that the local start gate time for Meas 1 is 3 ns.

MEASUREMENT:MEAS<x>:GLOBALref

This command sets or queries the reference levels global flag for the measurement.
Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:GLOBALref {OFF|ON|0|1}
MEASUREMENT:MEAS<x>:GLOBALref?

Arguments MEAS<x> specifies the measurement number.

OFF allows ref levels to be set separately for each measurement.

ON applies the same ref levels to all measurements.

0 allows ref levels to be set separately for each measurement.

1 applies the same ref levels to all measurements.

Examples `MEASUREMENT:MEAS1:GLOBALref 0` allows ref levels to be set separately for each measurement.

`MEASUREMENT:MEAS1:GLOBALref?` might return
`:MEASUREMENT:MEAS1:GLOBALREF 1` indicating the same ref levels apply to all measurements.

MEASUREMENT:MEAS<x>:HIGHREFVOLTAGE

This command sets or queries the high reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:HIGHREFVOLTAGE <NR3>`
`MEASUREMENT:MEAS<x>:HIGHREFVOLTAGE?`

Arguments `MEAS<x>` specifies the measurement number.
`<NR3>` is the high reference voltage value for the selected configuration.

Examples `MEASUREMENT:MEAS1:HIGHREFVOLTAGE 1.5` sets the high reference voltage to 1.5 V.

`MEASUREMENT:MEAS1:HIGHREFVOLTAGE?` might return
`:MEASUREMENT:MEAS1:HIGHREFVOLTAGE 1.0000` indicating the high reference voltage is set to 1.0 V.

MEASUREMENT:MEAS<x>:IDLETIME

This command sets or queries the idle time for the measurement when the measurement type is burst width. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:IDLETIME <NR3>`
`MEASUREMENT:MEAS<x>:IDLETIME?`

Arguments MEAS<x> specifies the measurement number.
<NR3> is the idle time.

Examples MEASUREMENT:MEAS1:IDLETime 40.0E-6 sets the idle time to 40.0 μ s.
MEASUREMENT:MEAS1:IDLETime? might return
:MEASUREMENT:MEAS1:IDLETIME 50.0000E-6 indicating the idle time
is 50.0 μ s.

MEASUREMENT:MEAS<x>:LABEL

This command sets or queries the label for the measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The measurement number is specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:LABEL <QString>

Arguments MEAS<x> specifies the measurement number.
<QString> is the measurement label.

Examples MEASUREMENT:MEAS1:LABEL "Delay" sets the label to Delay.
MEASUREMENT:MEAS1:LABEL? might return :MEASUREMENT:MEAS1:LABEL
"Peak-to-Peak" indicating that the measurement 1 label is Peak-to-peak.

MEASUREMENT:MEAS<x>:LOWREFVoltage

This command sets or queries the low reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:LOWREFVoltage <NR3>
MEASUREMENT:MEAS<x>:LOWREFVoltage?

Arguments MEAS<x> specifies the measurement number.
<NR3> is the low reference voltage value for the selected configuration.

Examples `MEASUREMENT:MEAS1:LOWREFVOLTAGE -1.30` sets the low reference voltage to -1.3 V.

`MEASUREMENT:MEAS1:LOWREFVOLTAGE?` might return
`:MEASUREMENT:MEAS1:LOWREFVOLTAGE -1.0000` indicating the low ref voltage is -1.0 V.

MEASUREMENT:MEAS<x>:PASSFAILENabled

This command returns or sets the pass/fail test enable status. If enabled, this will turn on pass fail testing for the specified measurement. Measurements are specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:PASSFAILENabled <NR1>`
`MEASUREMENT:MEAS<x>:PASSFAILENabled?`

Related Commands [ACTONEVENT:ENable](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEWAVEFORM:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE](#)

Arguments <NR1> enables or disables pass fail testing for the specified measurement. A value of 1 enables and a value of 0 disables.

Examples `MEASUREMENT:MEAS1:PASSFAILENabled 1` enables pass fail testing for the specified measurement.

`MEASUREMENT:MEAS1:PASSFAILENabled?` might return
`:MEASUREMENT:MEAS1:PASSFAILENabled 0` indicating that pass fail testing for the specified measurement is disabled.

MEASUREMENT:MEAS<x>:PASSFAILHIGHlimit

This command returns or sets the high limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MEAS<x>:PASSFAILHIGHlimit <NR2> MEASUREMENT:MEAS<x>:PASSFAILHIGHlimit?</pre>
Related Commands	<p>ACTONEVENT:ENABLE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEMODE:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE</p>
Arguments	<NR2> sets the high limit for a measurement test. The high limit is a number which a measurement result will be tested against.
Examples	<pre>MEASUREMENT:MEAS1:PASSFAILHIGHlimit 710</pre> <p>specifies 710 as the high limit for measurement 1.</p> <pre>MEASUREMENT:MEAS1:PASSFAILHIGHlimit?</pre> <p>might return</p> <pre>:MEASUREMENT:MEAS1:PASSFAILHIGHlimit 710</pre> <p>indicating that 710 is the high limit for measurement 1.</p>

MEASUREMENT:MEAS<x>:PASSFAILLIMIT

This command returns or sets the limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group	Measurement
Syntax	<pre>MEASUREMENT:MEAS<x>:PASSFAILLIMIT <NR2> MEASUREMENT:MEAS<x>:PASSFAILLIMIT?</pre>
Related Commands	<p>ACTONEVENT:ENABLE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEMODE:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE</p>

Arguments <NR2> sets the limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples MEASUREMENT:MEAS1:PASSFAILLIMIT 710 specifies 710 as the limit for measurement 1.

MEASUREMENT:MEAS1:PASSFAILLIMIT? might return
:MEASUREMENT:MEAS1:PASSFAILLIMIT 710 indicating that 710 is
the limit for measurement 1.

MEASUREMENT:MEAS<x>:PASSFAILLIMIT

This command returns or sets the low limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:PASSFAILLIMIT <NR2>
MEASUREMENT:MEAS<x>:PASSFAILLIMIT?

Related Commands [ACTONEVENT:ENABLE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEMODE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE](#)

Arguments <NR2> sets the low limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples MEASUREMENT:MEAS1:PASSFAILLIMIT 690 specifies 690 as the limit for measurement 1.

MEASUREMENT:MEAS1:PASSFAILLIMIT? might return
:MEASUREMENT:MEAS1:PASSFAILLIMIT 690 indicating that 690 is the
low limit for measurement 1.

MEASUREMENT:MEAS<x>:PASSFAILMARGIN

This command returns or sets the allowed margin for limit comparisons for all pass/fail checks. This is given as a percentage with a default value of 0.05 representing 5%. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:PASSFAILMARGIN <NR2>
MEASUREMENT:MEAS<x>:PASSFAILMARGIN?

Related Commands [ACTONEVENT:ENABLE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SAVEWAVEFORM:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:SRQ:STATE](#)
[ACTONEVENT:MEASUREMENT:ACTION:STOPACQ:STATE](#)

Arguments <NR2> sets the allowed margin for limit comparisons for all pass/fail checks. The margin as a percentage of the limit.

Examples MEASUREMENT:MEAS1:PASSFAILMARGIN 0.10 specifies 10% as the allowed margin for limit comparisons for all pass/fail checks.

MEASUREMENT:MEAS1:PASSFAILMARGIN? might return
:MEASUREMENT:MEAS1:PASSFAILMARGIN 0.10 indicating that 10% is the allowed margin for limit comparisons for all pass/fail checks on measurement 1.

MEASUREMENT:MEAS<x>:PASSFAILWHEN

This command sets or returns the condition on which a measurement test fails. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:PASSFAILWHEN {LESSthan| GREATERthan|
EQUALS| NOTEQUALS| INSIDERANGE| OUTSIDERANGE}
MEASUREMENT:MEAS<x>:PASSFAILWHEN?

Related Commands [ACTONEVent:ENable](#)
[ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE](#)
[ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE](#)
[ACTONEVent:MEASUrement:ACTION:SRQ:STATE](#)
[ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE](#)

Arguments **LESSthan** sets the condition for measurement test failure as less than the given limit. This is the default value.
GREATERthan sets the condition for measurement test failure as greater than the given limit.
Equals sets the condition for measurement test failure as equals the given limit.
NOTEQuals sets the condition for measurement test failure as not equal to the given limit.
INSIDErange sets the condition for measurement test failure as inside the limit range.
OUTSIDErange sets the condition for measurement test failure as outside the limit range.

Examples `MEASUrement:MEAS1:PASSFAILWHEN LESSthan` specifies the condition for measurement 1 test failure as less than the given limit.
`MEASUrement:MEAS1:PASSFAILWHEN?` might return
`:MEASUrement:MEAS1:PASSFAILWHEN LESSthan` indicating that the condition for measurement 1 test failure as less than the given limit.

MEASUrement:MEAS<x>:PERFREQ:EDGE

This command sets or queries the edge type of a Period/Frequency measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:PERFREQ:EDGE {FIRST|RISE|FALL}`

Arguments **MEAS<x>** specifies the measurement number.
FIRST computes the measurement between Rising edges if the first edge is Rising. Computes the measurement between Falling edges if the first edge is Falling.

RISE computes the measurement between Rising edges.

FALL computes the measurement between Falling edges.

Examples MEASUREMENT:MEAS1:PERFREQ:EDGE RISE specifies computing the measurement between Rising edges.

measurement:meas1:perfreq:edge? might return
:MEASUREMENT:MEAS1:PERFREQ:EDGE FIRST if the measurement is computed between edges of the first type edge found.

MEASUREMENT:MEAS<x>:POLARITY

This command sets or queries the polarity for the measurement when the measurement type is burst width. Measurements are specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:POLARITY {NORMAL|INVERTED}
MEASUREMENT:MEAS<x>:POLARITY?

Arguments MEAS<x> specifies the measurement number.

NORMAL specifies normal polarity.

INVERTED specifies inverted polarity.

Examples MEASUREMENT:MEAS1:POLARITY INVERTED specifies inverted polarity.

MEASUREMENT:MEAS1:POLARITY? might return
:MEASUREMENT:MEAS1:POLARITY NORMAL indicating normal polarity.

MEASUREMENT:MEAS<x>:REFLEVELS:ABSOLUTE:FALLHIGH

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. *This command affects the results of rise and fall measurements*

Group Measurement

Syntax MEASUREMENT:MEAS<x>:REFLEVELS:ABSOLUTE:FALLHIGH <NR3>
MEASUREMENT:MEAS<x>:REFLEVELS:ABSOLUTE:FALLHIGH?

Arguments MEAS<x> specifies the measurement number.
<NR3> is the high reference level in volts. The default is 0.0 V.

Examples MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.71 sets the high reference level to 1.71 V.

MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH? might return
:MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.7100E+00,
indicating that the absolute high reference level is set to 1.71 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLLOW

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. This command affects the results of rise and fall measurements.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLLOW <NR3>
MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLLOW?

Arguments MEAS<x> specifies the measurement number.
<NR3> is the low reference level in volts. The default is 0.0 V.

Examples MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0 V sets the low reference level to 0.0 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW? might return
:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0000E+00,
indicating that the absolute low reference level is set to 0.0 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLMID

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. *This command affects the results of period, frequency, delay, and all cyclic measurements.*

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLMID <NR3> MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:FALLMID?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the mid reference level in volts. The default is 0.0 V.
Examples	MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5 sets the mid reference level for the delay waveform to 0.5 V. MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5000E+00, indicating that the absolute mid reference level is set to 0.5 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:HYSTeresis <NR3> MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:HYSTeresis?
Arguments	MEAS<x> specifies the measurement number. <NR3> is the hysteresis value used for autoset.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS 25.0E-3` sets the hysteresis to 25 mV.

`MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30 mV.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEHIGH

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEHIGH <NR3>`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEHIGH?`

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the high reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH 1.50` sets the high reference level to 1.5 V.

`MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH 1.0000` indicating the high reference level is set to 1.0 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISELOW

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISELOW <NR3>`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISELOW?`

- Arguments** MEAS<x> specifies the measurement number.
- <NR3> is the low reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.
- Examples** MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW -1.50 sets the low reference level to -1.5 V.
- MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW -1.0000 indicating the low reference level is -1.0 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEMID

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

- Group** Measurement
- Syntax** MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEMID <NR3>
MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:RISEMID?
- Arguments** MEAS<x> specifies the measurement number.
- <NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.
- Examples** MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID 30.0E-3 sets the mid reference level to 30 mV.
- MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference level is 0.0 V.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:TYPE

This command sets or queries the reference level type for the measurement. The measurement number is specified by x.

- Group** Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:TYPE {SAME|UNIQUE}`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:ABSOLUTE:TYPE?`

Arguments `MEAS<x>` specifies the measurement number.

`SAME` specifies that the absolute levels are set the same.

`UNIQUE` specifies that the absolute levels can be set independently.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:TYPE?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:TYPE SAME` indicating the absolute levels are set the same.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:BASETOP

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The measurement number is specified by *x*.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:BASETOP`
`{AUTO|MINMAX|MEANHISTOGRAM|MODEHISTOGRAM|EYEHISTOGRAM}`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:BASETOP?`

Arguments `MEAS<x>` specifies the measurement number.

`AUTO` automatically chooses a reference level method.

`MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

`MEANHISTOGRAM` specifies that reference levels are relative to the histogram mean BASE and TOP.

`MODEHISTOGRAM` specifies that reference levels are relative to the histogram mode BASE and TOP.

`EYEHISTOGRAM` specifies that reference levels are relative to the eye histogram BASE and TOP.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:BASETOP MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

MEASUREMENT:MEAS1:REFLEVELS1:BASETOP? might return
:MEASUREMENT:MEAS1:REFLEVELS1:BASETOP AUTO indicating the
instrument automatically chooses a reference level method.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:METHOD

This command sets or queries the method used to calculate reference levels for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFLEVELS<x>:METHOD {PERCENT ABSOLUTE} MEASUREMENT:MEAS<x>:REFLEVELS<x>:METHOD?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>PERCENT specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT commands.</p> <p>ABSOLUTE specifies that the reference levels are set explicitly using the MEASUREMENT:MEAS<x>:REFLEVEL:ABSOLUTE commands. This method is useful when precise values are required.</p>
Examples	<p>MEASUREMENT:MEAS1:REFLEVELS1:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.</p> <p>MEASUREMENT:MEAS1:REFLEVELS1:METHOD? might return :MEASUREMENT:MEAS1:REFLEVELS1:METHOD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.</p>

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLHIGH <NR3> MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLHIGH?

Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level.</p>
Examples	<p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 95.0 sets the high reference level of the falling edge to 95%.</p> <p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 90.0000 indicating the high reference level is set to 90%.</p>

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLLOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLLOW <NR3> MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLLOW?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level.</p>
Examples	<p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW 5.0 sets the low reference level of the falling edge is 5%.</p> <p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW 10.0000 indicating the low reference level of the falling edge is 10%.</p>

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:FALLMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group	Measurement
Syntax	<pre>MEASurement:MEAS<x>:REFLevels<x>:PERCent:FALLMid <NR3> MEASurement:MEAS<x>:REFLevels<x>:PERCent:FALLMid?</pre>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level.</p>
Examples	<p>MEASurement:MEAS1:REFLevels1:PERCent:FALLMid 50.0 sets the mid reference level of the falling edge to 50%.</p> <p>MEASurement:MEAS1:REFLevels1:PERCent:FALLMid? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLMID 50.0000 indicating the mid reference level of the falling edge is set to 50%.</p>

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:HYSTERESIS

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group	Measurement
Syntax	<pre>MEASurement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis <NR3> MEASurement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis?</pre>
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p><NR3> is the hysteresis value used for the autoset.</p>
Examples	<p>MEASurement:MEAS1:REFLevels1:PERCent:HYSTeresis 5.0 sets the hysteresis level to 5%.</p> <p>MEASurement:MEAS1:REFLevels1:PERCent:HYSTeresis? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is 5.0%.</p>

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEHIGH <NR3>
MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEHIGH?

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH 95.0 sets the high reference level of the rising edge is set to 95%.

MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH? might return
:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH 90.0000
indicating the high reference level of the rising edge is set to 90%.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISELOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISELOW <NR3>
MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISELOW?

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW 5.0` sets the low reference level is set to 5.0%.

`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW 10.0000` indicating the low reference level is set to 10.0%.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEMID <NR3>`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:RISEMID?`

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples `MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID 50.0000` sets the mid reference level of the rising edge is set to 50.0%.

`MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID 50.0000` indicating the mid reference level of the rising edge is set to 50.0%.

MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:TYPE

This command specifies or queries the reference level percent type for the measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:TYPE`
`{TENNINETY|TWENTYEIGHTY|CUSTOM}`
`MEASUREMENT:MEAS<x>:REFLEVELS<x>:PERCENT:TYPE?`

Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>TENNInety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.</p> <p>TWENTyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.</p> <p>CUSTOM allows setting other reference level percents.</p>
Examples	<p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE TWENTyeighty sets the reference levels percent to 20%, 50% and 80%.</p> <p>MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE TENNINETY indicating the reference levels percent type is 10%, 50% and 90% respectively</p>

MEASUREMENT:MEAS<x>:REFMODE

This command sets or queries the reference level mode for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFMODE {AUTO MANUAL} MEASUREMENT:MEAS<x>:REFMODE?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>AUTO sets the reference level for the measurement automatically.</p> <p>MANUAL allows the user to set the reference level for the measurement.</p>
Examples	<p>MEASUREMENT:MEAS1:REFMODE MANUAL allows the user to set the reference level for the measurement.</p> <p>MEASUREMENT:MEAS1:REFMODE? might return :MEASUREMENT:MEAS1:REFMODE AUTO indicating the reference levels for the measurement are set automatically.</p>

MEASUREMENT:MEAS<x>:REFVOLTAGE

This command sets or queries the reference voltage value for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:REFVOLTAGE <NR3> MEASUREMENT:MEAS<x>:REFVOLTAGE?
Arguments	<NR3> is the reference voltage value for the selected configuration.
Examples	MEASUREMENT:MEAS1:REFVOLTAGE 30.E-3 sets the reference voltage value for the measurement to 30 mV. MEASUREMENT:MEAS1:REFVOLTAGE? might return :MEASUREMENT:MEAS1:REFVOLTAGE 0.0E+0 indicating the reference voltage value for the measurement is 0.0 V.

MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MAXIMUM? (Query Only)

This query-only command returns the maximum value for all accumulated measurement acquisitions of the specified measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MAXIMUM?
Returns	The maximum value for all accumulated measurement acquisitions of the specified measurement.
Examples	MEASUREMENT:MEAS1:RESULTS:ALLACQS:MAXIMUM? might return :MEASUREMENT:MEAS1:RESULTS:ALLACQS:MAXIMUM 2.420 indicating the maximum measurement value is 2.420 V.

MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MEAN? (Query Only)

This query-only command returns the mean value for all accumulated measurement acquisitions for measurement <x>.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MEAN?

Returns The mean value for all accumulated measurement acquisitions for measurement <x>.

Examples MEASUREMENT:MEAS1:RESULTS:ALLACQS:MEAN? might return
:MEASUREMENT:MEAS1:RESULTS:ALLACQS:MEAN 2.2807617754647
indicating the measurement mean is 2.28 V.

MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MINIMUM? (Query Only)

This query-only command returns the minimum value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:MINIMUM?

Returns The minimum value for all accumulated measurement acquisitions for measurement <x>.

Examples MEASUREMENT:MEAS1:RESULTS:ALLACQS:MINIMUM? might return
:MEASUREMENT:MEAS1:RESULTS:ALLACQS:MINIMUM 2.220 indicating the
measurement minimum value is 2.220 V.

MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:RESULTS:ALLACQS:PK2PK?

Returns The peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Examples MEASUREMENT:MEAS1:RESULTS:ALLACQS:PK2PK? might return
:MEASUREMENT:MEAS1:RESULTS:ALLACQS:PK2PK 200.0E-3 indicating the
measurement peak-to-peak value is 200 mV.

MEASUREMENT:MEAS<x>:RESULTS:ALLAcqs:POPULATION? (Query Only)

This query-only command returns the population measurement value for measurement <x>.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:RESULTS:ALLAcqs:POPULATION?
Returns	The population measurement value for measurement <x>.
Examples	MEASUREMENT:MEAS1:RESULTS:ALLAcqs:POPULATION? might return :MEASUREMENT:MEAS1:RESULTS:ALLAcqs:POPULATION 29825 indicating the measurement population is 29825 measurements.

MEASUREMENT:MEAS<x>:RESULTS:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation for all accumulated measurement acquisitions for measurement <x>.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:RESULTS:ALLAcqs:STDDev?
Returns	The standard deviation for all accumulated measurement acquisitions for measurement <x>.
Examples	MEASUREMENT:MEAS1:RESULTS:ALLAcqs:STDDev? might return :MEASUREMENT:MEAS1:RESULTS:ALLAcqs:STDDev 23.5741246494459E-3 indicating the measurement standard deviation is 23.57 mV.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTAcq:MAXimum? (Query Only)

This query-only command returns the maximum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group	Measurement
--------------	-------------

Syntax	<code>MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:MAXIMUM?</code>
Returns	The maximum value found for the specified measurement since the last statistical reset.
Examples	<code>MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM?</code> might return <code>:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM 2.280</code> indicating the current measurement maximum value is 2.28 V.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:MEAN? (Query Only)

This query-only command returns the mean value for the measurement for the current acquisition.

Group	Measurement
Syntax	<code>MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:MEAN?</code>
Arguments	<code>MEAS<x></code> is the measurement number for which to return a value. This is the equivalent of the number shown in the measurement badge on the UI.
Returns	The mean value accumulated for the specified measurement since the last statistical reset.
Examples	<code>MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN?</code> might return <code>:MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN 2.30</code> indicating the mean of measurement 3 in the current acquisition is 2.3 V.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:MINIMUM? (Query Only)

This query-only command returns the minimum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group	Measurement
Syntax	<code>MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:MINIMUM?</code>

Returns The minimum value found for the specified measurement since the last statistical reset.

Examples `MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MINIMUM?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MINIMUM 2.260` indicating the minimum value of the measurement in the current acquisition is 2.26 V.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:PK2PK?`

Returns The peak-to-peak value for the specified measurement.

Examples `MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:PK2PK?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:PK2PK 0.0E+0` indicating the peak-to-peak value for the specified measurement for the current acquisition is 0.0 V.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:POPULATION? (Query Only)

This query-only command returns the population for the specified measurement for the current acquisition. The measurement number is specified by x.

Group Measurement

Syntax `MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:POPULATION?`

Returns The population measurement value for the specified measurement.

Examples `MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:POPULATION?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:POPULATION 1` indicating the population for the current measurement is 1 measurement.

MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:STDDEV? (Query Only)

This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:STDDEV?
Returns	The standard deviation of values accumulated for the specified measurement since the last statistical reset.
Examples	MEASUREMENT:MEAS<x>:RESULTS:CURRENTACQ:STDDEV? might return :MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:STDDEV 0.0E+0 indicating the standard deviation for the measurement is 0.0 V.

MEASUREMENT:MEAS<x>:SIGNALTYPE

This command sets or queries the signal type of source 1 for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:SIGNALTYPE {CLOCK DATA AUTO} MEASUREMENT:MEAS<x>:SIGNALTYPE?
Arguments	MEAS<x> specifies the measurement number. CLOCK specifies a clock signal type. DATA specifies a data signal type. AUTO automatically selects the signal type.
Examples	MEASUREMENT:MEAS1:SIGNALTYPE CLOCK specifies a clock signal type. MEASUREMENT:MEAS1:SIGNALTYPE? might return :MEASUREMENT:MEAS1:SIGNALTYPE AUTO indicating the instrument automatically selects the signal type.

MEASUREMENT:MEAS<x>:SOURCE<x>

This command sets or queries the measurement source. The measurement number and source are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:SOURCE<x> {CH<x> DCH<x>_D<x> MATH<x> REF<x>} MEASUREMENT:MEAS<x>:SOURCE<x>?
Arguments	<p>MEAS<x> specifies the measurement number.</p> <p>SOURCE<x> specifies the source number.</p> <p>CH<x> specifies an analog channel to use as the source.</p> <p>DCH<x>_D<x> specifies a digital channel to use as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.</p> <p>MATH<x> specifies a math waveform to use as the source.</p> <p>REF<x> specifies a reference waveform to use as the source.</p>
Examples	<p>MEASUREMENT:MEAS1:SOURCE1 CH1 sets the source for measurement 1 to channel 1.</p> <p>MEASUREMENT:MEAS1:SOURCE1? might return :MEASUREMENT:MEAS1:SOURCE CH1 indicating the measurement source is channel 1.</p>

MEASUREMENT:MEAS<x>:STATUS? (Query Only)

This command returns the pass fail status, if applicable, for the selected measurement. Measurements are specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:STATUS?
Related Commands	<p>ACTONEVENT:ENABLE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEIMAGE:STATE</p> <p>ACTONEVENT:MEASUREMENT:ACTION:SAVEMATHFORM:STATE</p>

`ACTONEVent:MEASUrement:ACTION:SRQ:STATE`

`ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE`

Arguments PASS specifies that the user specified measurement limit has not been violated.
FAIL specifies that the user specified measurement limit has been violated.

Examples `MEASUrement:MEAS1:STATUS?` might return `:MEASUrement:MEAS1:STATUS`
PASS indicating that the user specified measurement limit has not been violated.

MEASUrement:MEAS<x>:TOEdge

This command sets or queries the 'to edge' type for the measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:TOEdge`
{`SAMEas`|`OPPOSITEas`|`RISE`|`FALL`|`BOTH`}
`MEASUrement:MEAS<x>:TOEdge?`

Arguments `MEAS<x>` specifies the measurement number.
`FALL` specifies the falling edge of the waveform.
`RISE` specifies the rising edge of the waveform.
`BOTH` specifies both a rising and falling edge of the waveform.
`SAMEas` specifies that both edges of the waveform are the same.
`OPPOSITEas` specifies that the edges of the waveform are not the same.

Examples `MEASUrement:MEAS1:TOEdge FALL` specifies the to edge is the falling edge of the waveform.
`MEASUrement:MEAS1:TOEdge?` might return `:MEASUrement:MEAS1:TOEDGE`
`SAMEAS` indicating that both edges of the waveform are the same.

MEASUrement:MEAS<x>:TOEDGESEARCHDIRect

This command sets or queries the to edge search direction for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:TOEDGESEARCHDIRECT {FORWARD BACKWARD} MEASUREMENT:MEAS<x>:TOEDGESEARCHDIRECT?
Arguments	MEAS<x> specifies the measurement number. FORWARD specifies a forward search to the edge. BACKWARD specifies a backward search to the edge.
Examples	MEASUREMENT:MEAS1:TOEDGESEARCHDIRECT FORWARD specifies a forward search to the edge. MEASUREMENT:MEAS1:TOEDGESEARCHDIRECT? might return :MEASUREMENT:MEAS1:TOEDGESEARCHDIRECT FORWARD indicating the instrument will search in the forward direction for the to edge.

MEASUREMENT:MEAS<x>:TRANSITION

This command sets or queries the transition edges flag for the measurement. The measurement number is specified by x.

Group	Measurement
Syntax	MEASUREMENT:MEAS<x>:TRANSITION {<NR1> OFF ON}
Arguments	<NR1> = 1, the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only. If it is set to 0, the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges. OFF computes the measurement on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges. ON computes the measurement on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.
Examples	MEASUREMENT:MEAS1:TRANSITION 1 specifying the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only. MEASUREMENT:MEAS1:TRANSITION? might return :MEASUREMENT:MEAS1:TRANSITION 0 indicating the measurement is

computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

MEASUREMENT:MEAS<x>:TYPE

This command sets or queries the measurement type for the measurement specified by x.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:TYPE {ACRMS |AMPLITUDE |AREA |BASE |BURSTWIDTH |DATARATE |DELAY |FALLSLEWRATE |FALLTIME |FREQUENCY |HIGHTIME |HOLD |LOWTIME |MAXIMUM |MEAN |MINIMUM |NDUTY |NPERIOD |NOVERSHOOT |NWIDTH |PDUTY |PERIOD |PHASE |PK2PK |POVERSHOOT |PWIDTH |RISESLEWRATE |RISETIME |RMS |SETUP |SKEW |TIMEOUTSIDELEVEL |TOP}
MEASUREMENT:MEAS<x>:TYPE?

Arguments ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

$$\textit{Amplitude} = \textit{High} - \textit{Low}$$

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DELAY is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQUENCY is the reciprocal of Period. This measurement is made on each cycle in the record.

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXIMUM is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINIMUM is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDUTY (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NOVERSHOOT (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

$$\text{Negative Overshoot} = (\text{Base} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

NWIDTH (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{Top}) / \text{Amplitude} \times 100\%$$

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

Examples

MEASUREMENT:MEAS2:TYPE FREQUENCY defines measurement 2 as a measurement of the frequency of a waveform.

MEASUREMENT:MEAS1:TYPE? might return **:MEASUREMENT:MEAS1:TYPE RMS**, indicating that measurement 1 is defined to measure the RMS value of a waveform.

MEASUREMENT:MEAS<x>:XUNIT? (Query Only)

Returns the horizontal scale units of the specified measurement.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:XUNIT?

Arguments MEAS<x> specifies the measurement number.

Examples MEASUREMENT:MEAS2:XUNIT? might return "V", indicating that the horizontal unit for measurement 2 is seconds.

MEASUREMENT:MEAS<x>:YUNIT? (Query Only)

Returns the vertical scale units of the specified measurement.

Group Measurement

Syntax MEASUREMENT:MEAS<x>:YUNIT?

Arguments MEAS<x> specifies the measurement number.

Examples MEASUREMENT:MEAS12:YUNIT? might return "V", indicating that the vertical unit for measurement 12 is volts.

MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH <NR3>
MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH?

Arguments	<NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.
Examples	<code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.5</code> sets the high reference level to 1.5 V. <code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH?</code> might return <code>:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000</code> indicating the high reference level is 1.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group	Measurement
Syntax	<code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW <NR3></code> <code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW?</code>
Arguments	<NR3> is the value used as the low reference level of the falling edge.
Examples	<code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.5</code> sets the low reference level to -1.5 V. <code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW?</code> might return <code>:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.0000</code> indicating the low reference level is -1.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group	Measurement
Syntax	<code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID <NR3></code> <code>MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID?</code>
Arguments	<NR3> is the value used as the mid reference level of the falling edge.

Examples `MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID 10.0E-3` sets the mid reference level to 10.0 mV.

`MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID?` might return
`:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating the mid reference level is 0.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS <NR3>`
`MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS?`

Arguments `<NR3>` is the value of the hysteresis of the reference level.

Examples `MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS?` might return
`:MEASUREMENT:REFLEVELS:ABSOLUTE:HYS TERESIS 30.0000E-3`
 indicating the hysteresis is set to 30.0 mV.

MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH <NR3>`
`MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH?`

Arguments `<NR3>` is the value used as the high reference level of the rising edge.

Examples `MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH 1.5` sets the high reference to 1.5 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH? might return
:MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the
high reference is 1.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW <NR3> MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW?
Arguments	<NR3> is the value used as the the low reference level of the rising edge
Examples	MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW -1.5 sets the low reference level to -1.5 V. MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW -1.0000 indicating the low reference level is -1.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID <NR3> MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID?
Arguments	<NR3> is the mid reference level of the rising edge.
Examples	MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID 10.0E-3 sets the mid reference to 10.0 mV.

MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID? might return
:MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the mid
reference is 0.0 V.

MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE

This command sets or queries the reference level type for the measurement.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE {SAME UNIQUE} MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE?
Arguments	SAME specifies that the absolute levels are set the same. UNIQUE specifies that the absolute levels can be set independently.
Examples	MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE UNIQUE specifies that the absolute levels can be set independently. MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE SAME indicating the absolute levels are set the same.

MEASUREMENT:REFLEVELS:BASETOP

This command sets or queries the method used to calculate the TOP and BASE,
used to calculate reference levels for the measurement.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:BASETOP {AUTO MINMAX MEANHISTOGRAM MODEHISTOGRAM EYEHISTOGRAM} MEASUREMENT:REFLEVELS:BASETOP?
Arguments	Arguments are the base top methods.
Examples	MEASUREMENT:REFLEVELS:BASETOP MINMAX selects the MINMAX base top method.

MEASUREMENT:REFLEVELS:BASETOP? might return
:MEASUREMENT:REFLEVELS:BASETOP MINMAX indicating the base top method
is MINMAX.

MEASUREMENT:REFLEVELS:METHOd

This command sets or queries the method used to calculate reference levels for the measurement.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:METHOd {PERCENT ABSOLUTE} MEASUREMENT:REFLEVELS:METHOd?
Arguments	<p>PERCENT specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:REFLEVELS:REFLEVEL:PERCENT commands.</p> <p>ABSOLUTE specifies that the reference levels are set explicitly using the MEASUREMENT:REFLEVELS:REFLEVEL:ABSOLUTE commands. This method is useful when precise values are required.</p>
Examples	<p>MEASUREMENT:REFLEVELS:METHOd ABSOLUTE specifies that the reference levels are set explicitly.</p> <p>MEASUREMENT:REFLEVELS:METHOd? might return :MEASUREMENT:REFLEVELS:METHOd PERCENT indicating the reference levels are calculated as a percent relative to HIGH and LOW.</p>

MEASUREMENT:REFLEVELS:MODE

This command sets or queries how often reference levels are calculated.

Group	Measurement
Syntax	MEASUREMENT:REFLEVELS:MODE {LATCH CONTINUOUS} MEASUREMENT:REFLEVELS:MODE?
Arguments	<p>LATCH calculates reference levels only on the first acquisition after a statistics reset.</p> <p>CONTINUOUS calculates reference levels on every acquisition.</p>

Examples `MEASUREMENT:REFLEVELS:MODE CONTINUOUS` calculates reference levels on every acquisition.

`MEASUREMENT:REFLEVELS:MODE?` might return
`:MEASUREMENT:REFLEVELS:MODE LATCH` indicating reference levels are calculated only on the first acquisition after a statistics reset.

MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH <NR3>`
`MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH?`

Arguments `<NR3>` is the percentage used to calculate the high reference level of the falling edge

Examples `MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 90` sets the high reference level to 90%.

`MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH?` might return
`:MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 80.0000` indicating the high reference level is 80%.

MEASUREMENT:REFLEVELS:PERCENT:FALLLOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:FALLLOW <NR3>`
`MEASUREMENT:REFLEVELS:PERCENT:FALLLOW?`

Arguments `<NR3>` is the percentage used to calculate the mid reference level of the falling edge.

Examples `MEASUREMENT:REFLEVELS:PERCENT:FALLLOW 10` sets the low reference level to 10%.

`MEASUREMENT:REFLEVELS:PERCENT:FALLLOW?` might return
 `:MEASUREMENT:REFLEVELS:PERCENT:FALLLOW 20.0000` indicating the low reference level is 20%.

MEASUREMENT:REFLEVELS:PERCENT:FALLMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:FALLMID <NR3>`
 `MEASUREMENT:REFLEVELS:PERCENT:FALLMID?`

Arguments `<NR3>` is the percentage used to calculate the mid reference level of the falling edge

Examples `MEASUREMENT:REFLEVELS:PERCENT:FALLMID 55` sets the mid reference level to 55%.

`MEASUREMENT:REFLEVELS:PERCENT:FALLMID?` might return
 `:MEASUREMENT:REFLEVELS:PERCENT:FALLMID 50.0000` indicating the mid reference level is 50%.

MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS <NR3>`
 `MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS?`

Arguments `<NR3>` is the percentage used to calculate the hysteresis of the reference level.

Examples `MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS 3` sets the hysteresis to 3%.
`MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS?` might return
`:MEASUREMENT:REFLEVELS:PERCENT:HYS TERESIS 5.0000` indicating the
hysteresis is set to 5%.

MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH <NR3>`
`MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH?`

Arguments `<NR3>` is the percentage used to calculate the high reference level of the rising edge.

Examples `MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH 90` sets the high reference level to 90%.
`MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH?` might return
`:MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH 80.0000` indicating the
high ref level is 80%.

MEASUREMENT:REFLEVELS:PERCENT:RISELOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:RISELOW <NR3>`
`MEASUREMENT:REFLEVELS:PERCENT:RISELOW?`

Arguments `<NR3>` is the percentage used to calculate the low reference level of the rising edge.

Examples `MEASUREMENT:REFLEVELS:PERCENT:RISELOW 10` sets the low reference level to 10%.

`MEASUREMENT:REFLEVELS:PERCENT:RISELOW?` might return
`:MEASUREMENT:REFLEVELS:PERCENT:RISELOW 20.0000` indicating the low reference level is 20%.

MEASUREMENT:REFLEVELS:PERCENT:RISEMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:RISEMID <NR3>`
`MEASUREMENT:REFLEVELS:PERCENT:RISEMID?`

Arguments `<NR3>` is the percentage used to calculate the mid reference level of the rising edge.

Examples `MEASUREMENT:REFLEVELS:PERCENT:RISEMID 55` sets the mid reference level to 55%.

`MEASUREMENT:REFLEVELS:PERCENT:RISEMID?` might return
`:MEASUREMENT:REFLEVELS:PERCENT:RISEMID 50.0000` indicating the mid reference level is 50%.

MEASUREMENT:REFLEVELS:PERCENT:TYPE

This command sets or queries the reference level percent type for the measurement.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:PERCENT:TYPE`
`{TENNINETY|TWENTYEIGHTY|CUSTOM}`
`MEASUREMENT:REFLEVELS:PERCENT:TYPE?`

Arguments `TENNINETY` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTOM allows setting other reference level percents.

Examples `MEASUREMENT:REFLEVELS:PERCENT:TYPE TENNINETY` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`MEASUREMENT:REFLEVELS:PERCENT:TYPE?` might return `:MEASUREMENT:REFLEVELS:PERCENT:TYPE CUSTOM` indicating that custom reference levels can be set.

MEASUREMENT:REFLEVELS:TYPE

This command sets or queries the shared reference level method used for sources of measurement calculations.

Group Measurement

Syntax `MEASUREMENT:REFLEVELS:TYPE {GLOBAL|PERSOURCE}`
`MEASUREMENT:REFLEVELS:TYPE?`

Arguments `GLOBAL` shares reference levels across measurements.
`PERSOURCE` causes reference levels to be used on individual measurements.

Examples `MEASUREMENT:REFLEVELS:TYPE PERSOURCE` causes reference levels to be used on individual measurements.

`MEASUREMENT:REFLEVELS:TYPE?` might return `:MEASUREMENT:REFLEVELS:TYPE GLOBAL` indicating reference levels are shared across measurements.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLHIGH

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax `MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLHIGH <NR3>`
`MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLHIGH?`

Arguments <NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH 1.5 sets the reference level to 1.5 V.

MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH? might return
:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLLOW

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLLOW <NR3>
MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLLOW?

Arguments <NR3> is the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW -1.5 sets the low reference level to -1.5 V.

MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW? might return
:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the low reference level is -1.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMID

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMID <NR3>
MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:FALLMID?

Arguments <NR3> is the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID 50.0E-3` sets the mid reference level to 50.0 mV.
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID?` might return `:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating the reference level is 0.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYS TERESIS

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group Measurement

Syntax `MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYS TERESIS <NR3>`
`MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:HYS TERESIS?`

Arguments <NR3> is the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Examples `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYS TERESIS 20.0E-3` sets the hysteresis to 20.0 mV.
`MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYS TERESIS?` might return `:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYS TERESIS 30.0000E-3` indicating the hysteresis is 30.0 mV.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEHIGH

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax `MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEHIGH <NR3>`
`MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEHIGH?`

- Arguments** <NR3> is the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.
- Examples** `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH 1.5` sets the reference level to 1.5 V.
- `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH?` might return `:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the reference level is 1.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISELOW

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

- Group** Measurement
- Syntax** `MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISELOW <NR3>`
`MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISELOW?`
- Arguments** <NR3> is the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.
- Examples** `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISELOW -1.5` sets the reference level to -1.5 V.
- `MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISELOW?` might return `:MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISELOW -1.0000` indicating the reference level is -1.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEMID

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

- Group** Measurement
- Syntax** `MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEMID <NR3>`
`MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:RISEMID?`

Arguments <NR3> is the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEMID 10.0E-3 sets the reference level to 10.0 mV.
 MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEMID? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference level is 0.0 V.

MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:TYPE

This command sets or queries the reference level type for the measurement.

Group Measurement

Syntax MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:TYPE {SAME|UNIQUE}
 MEASUREMENT:REF<x>:REFLEVELS:ABSOLUTE:TYPE?

Arguments SAME specifies that the absolute levels are set the same.
 UNIQUE specifies that the absolute levels can be set independently.

Examples MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:TYPE UNIQUE specifies that the absolute levels can be set independently.
 MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:TYPE? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the absolute levels are set the same.

MEASUREMENT:REF<x>:REFLEVELS:BASETOP

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group Measurement

Syntax MEASUREMENT:REF<x>:REFLEVELS:BASETOP

Arguments	<p>AUTO automatically chooses a reference level method.</p> <p>MINMAX specifies that reference levels are relative to the measurement MIN and MAX.</p> <p>MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.</p> <p>MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.</p> <p>EYEHistogram specifies that reference levels are relative to the eye histogram BASE and TOP.</p>
Examples	<p>MEASUREMENT:REF1:REFLEVELS:BASETOP MINMAX specifies that reference levels are relative to the measurement MIN and MAX.</p> <p>MEASUREMENT:REF1:REFLEVELS:BASETOP? might return :MEASUREMENT:REF1:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.</p>

MEASUREMENT:REF<x>:REFLEVELS:METHOd

This command sets or queries the method used to calculate reference levels for the measurement.

Group	Measurement
Syntax	<p>MEASUREMENT:REF<x>:REFLEVELS:METHOd {PERCENT ABSOLUTE}</p> <p>MEASUREMENT:REF<x>:REFLEVELS:METHOd?</p>
Arguments	<p>PERCENT specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:REF<x>:REFLEVEL:PERCENT commands.</p> <p>ABSOLUTE specifies that the reference levels are set explicitly using the MEASUREMENT:REF<x>:REFLEVEL:ABSOLUTE commands. This method is useful when precise values are required.</p>
Examples	<p>MEASUREMENT:REF1:REFLEVELS:METHOd ABSOLUTE specifies that the reference levels are set explicitly.</p> <p>MEASUREMENT:REF1:REFLEVELS:METHOd? might return :MEASUREMENT:REF1:REFLEVELS:METHOd PERCENT indicating the reference levels are calculated as a percent relative to HIGH and LOW.</p>

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLHIGH <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLHIGH?
Arguments	<NR3> is the percentage used to calculate the high reference level of the falling edge
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH 95 sets the reference level to 95%. MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is 90%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLLOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLLOW <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLLOW?
Arguments	<NR3> is the percentage used to calculate the low reference level
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLLOW 5 sets the low reference level to 5%. MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLLOW? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the low reference level is 10%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLMID <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:FALLMID?
Arguments	<NR3> is the percentage used to calculate the mid reference level of the falling edge.
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID 55 sets the reference level to 50%. MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the mid reference level is 50%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:HYS TERESIS

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:HYS TERESIS <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:HYS TERESIS?
Arguments	<NR3> is the percentage used to calculate the hysteresis of the reference level.
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:HYS TERESIS 2 sets the hysteresis to 2%. MEASUREMENT:REF1:REFLEVELS:PERCENT:HYS TERESIS? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:HYS TERESIS 5.0000 indicating the hysteresis is 5%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEHIGH

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEHIGH <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEHIGH?
Arguments	<NR3> is the percentage used to calculate the high reference level of the rising edge
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH 95 sets the reference level to 95%. MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH 90.0000 indicating the reference level is 90%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISELOW

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISELOW <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISELOW?
Arguments	<NR3> is the percentage used to calculate the low reference level of the rising edge.
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:RISELOW 5 sets the low reference level to 5%. MEASUREMENT:REF1:REFLEVELS:PERCENT:RISELOW? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:RISELOW 10.0000 indicating the reference level is 10%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEMID

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEMID <NR3> MEASUREMENT:REF<x>:REFLEVELS:PERCENT:RISEMID?
Arguments	<NR3> is the percentage used to calculate the mid reference level of the rising edge.
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID 55 sets the mid reference level to 55%. MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the mid reference level is 50%.

MEASUREMENT:REF<x>:REFLEVELS:PERCENT:TYPE

This command sets or queries the reference level percent type for the measurement.

Group	Measurement
Syntax	MEASUREMENT:REF<x>:REFLEVELS:PERCENT:TYPE {TENNINETY TWENTYEIGHTY CUSTOM} MEASUREMENT:REF<x>:REFLEVELS:PERCENT:TYPE?
Arguments	TENNINETY sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively. TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively. CUSTOM allows setting other reference level percents.
Examples	MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE?` might return
`:MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE TENNINETY` indicating the
values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

MEASUREMENT:STATISTICS:CYCLEMODE

This command turns on and off cycle to cycle measurement statistics tracking and affects computation and display of cycle-cycle statistics in the Measurement Result table. It affects measurement statistics after being enabled and after new data is acquired and measured.

Group	Measurement
Syntax	<code>MEASUREMENT:STATISTICS:CYCLEMODE {OFF ON 0 1}</code> <code>MEASUREMENT:STATISTICS:CYCLEMODE?</code>
Arguments	<p><code>OFF</code> turns off statistics for all measurements. This is the default value.</p> <p><code>ON</code> turns on statistics and displays all statistics for each measurement.</p> <p><code>0</code> turns off statistics for all measurements.</p> <p><code>1</code> turns on statistics and displays all statistics for each measurement.</p>
Examples	<p><code>MEASUREMENT:STATISTICS:CYCLEMODE OFF</code> turns off statistics for all measurements.</p> <p><code>MEASUREMENT:STATISTICS:CYCLEMODE?</code> might return <code>:MEASUREMENT:STATISTICS:CYCLEMODE 1</code> indicating that statistics are displayed for each measurement.</p>

NEWpass (No Query Form)

This command (no query form) changes the password that enables access to password protected data. The `PASSWORD` command must be successfully executed before using this command or an execution error will be generated.

Group	Miscellaneous
Syntax	<code>NEWpass <QString></code>

Related Commands	PASSWord *PUD
Arguments	<Qstring> is the new password, which can contain up to 10 characters.
Examples	NEWPASS"mypassword" creates a new password (mypassword) for accessing your protected data.

*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The *OPC? query places the ASCII character “1” into the output queue when all such OPC commands are complete. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The *OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods. Refer to the *Oscilloscope operations that can generate OPC* table for a list of commands that generate an OPC message. (See Table 3-3.)

Group	Status and Error
Syntax	*OPC *OPC?
Related Commands	BUSY? *WAI
Examples	*OPC generates the operation complete message in the SESR at the completion of all pending OPC operations. *OPC? might return 1 to indicate that all pending OPC operations are finished.

*OPT? (Query Only)

This query-only command returns a comma separated list of installed options as an arbitrary ASCII string (no quotes) of the form:

<optionCode>:<optionDescription>,<optionCode>:<optionDescription>...

The last section of each entry (the text following the last hyphen) indicates the license type.

If no options are found, NONE is returned.

Group	Status and Error
Syntax	*OPT?
Examples	*OPT? response (with each option listed on a separate line for clarity): 2-ULTIMATE:License; Ultimate Bundle for 2 Series MSO, includes Source and Serial options; Node Locked, LIC2-500:License; 2 Series 500MHz bandwidth; Node Locked

PASSWord (No Query Form)

This command (no query form) enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the instrument is powered off, or until the FACTory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

Group	Miscellaneous
Syntax	PASSword <QString>
Related Commands	NEWpass *PUD
Arguments	<QString> is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.
Examples	PASSWORD "XYZZY" enables the *PUD and NEWPass set commands. PASSWORD disables the *PUD and NEWPass set commands. You can still use the query version of *PUD.

PAUSE (No Query Form)

This command causes the interface to pause the specified number of seconds before processing any other commands.

Group	Miscellaneous
Syntax	PAUSE <NR3>
Arguments	<NR3> is the specified number of seconds the interface is to pause before processing any other commands. The pause time is specified as a floating point value in seconds and must be > 0.0 and ≥ 1800.0 .
Examples	PAUSE 10.0e0; :ACQUIRE:NUMACQ causes the interface to pause 10 seconds before returning the number of acquisitions.

PG:AMPLitude

This command sets or queries the Pattern Generator output voltage. You can give any value, but the command will replace it with the nearest valid value (2.5, 3.3 and 5). In burst mode, only 5 V is supported.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	PG:AMPLitude {2.5 3.3 5V} PG:AMPLitude?
Arguments	2.5 sets the Pattern Generator amplitude to 2.5 V. 3.3 sets the Pattern Generator amplitude to 3.3 V. 5 sets the Pattern Generator amplitude to 5 V. This is the default value.
Examples	PG:AMPLitude 2.5 sets the Pattern Generator output voltage to 2.5 V. PG:AMPLitude? might return :PG:AMPLitude 5, indicating the Pattern Generator output voltage is 5 V.

PG:BIT:ONE

This command sets or queries the output value of Pattern Generator bit 1.

Conditions Requires option 2-SOURCE.

Group Pattern Generator

Syntax PG:BIT:ONE {HIGH|LOW|TOGGLE|HIGH_Z}
PG:BIT:ONE?

Arguments LOW sets the Pattern Generator bit value to Low.
HIGH sets the Pattern Generator bit value to High.
TOGGLE sets the Pattern Generator bit value to Toggle.
HIGH_Z sets the Pattern Generator bit value to Hi-Z. Only available for Continuous mode.

Examples PG:BIT:ONE TOGGLE sets the Pattern Generator bit 1 output to toggle state.
PG:BIT:ONE? might return :PG:BIT:ONE HIGH, indicating the Pattern Generator bit 1 output is set to High.

PG:BIT:THREE

This command sets or queries the output value of Pattern Generator bit 3.

Conditions Requires option 2-SOURCE.

Group Pattern Generator

Syntax PG:BIT:THREE {HIGH|LOW|TOGGLE|HIGH_Z}
PG:BIT:THREE?

Arguments LOW sets the Pattern Generator bit value to Low.
HIGH sets the Pattern Generator bit value to High.
TOGGLE sets the Pattern Generator bit value to Toggle.

HIGH_Z sets the Pattern Generator bit value to Hi-Z. Only available for Continuous mode.

Examples PG:BIT:THREE TOGGLE sets the Pattern Generator bit 3 output to toggle state.
PG:BIT:THREE? might return :PG:BIT:THREE HIGH, indicating the Pattern Generator bit 3 output is set to High.

PG:BIT:TWO

This command sets or queries the output value of Pattern Generator bit 2.

Conditions Requires option 2-SOURCE.

Group Pattern Generator

Syntax PG:BIT:TWO {HIGH|LOW|TOGGLE|HIGH_Z}
PG:BIT:TWO?

Arguments LOW sets the Pattern Generator bit value to Low.
HIGH sets the Pattern Generator bit value to High.
TOGGLE sets the Pattern Generator bit value to Toggle.
HIGH_Z sets the Pattern Generator bit value to Hi-Z. Only available for Continuous mode.

Examples PG:BIT:TWO TOGGLE sets the Pattern Generator bit 2 output to toggle state.
PG:BIT:TWO? might return :PG:BIT:TWO HIGH, indicating the Pattern Generator bit 2 output is set to High.

PG:BIT:ZERO

This command sets or queries the output value of Pattern Generator bit 0.

Conditions Requires option 2-SOURCE.

Group Pattern Generator

Syntax	<code>PG:BIT:ZERO {HIGH LOW TOGGLE HIGH_Z}</code> <code>PG:BIT:ZERO?</code>
Arguments	<p><code>LOW</code> sets the Pattern Generator bit value to Low.</p> <p><code>HIGH</code> sets the Pattern Generator bit value to High.</p> <p><code>TOGGLE</code> sets the Pattern Generator bit value to Toggle.</p> <p><code>HIGH_Z</code> sets the Pattern Generator bit value to Hi-Z. Only available for Continuous mode.</p>
Examples	<p><code>PG:BIT:ZERO TOGGLE</code> sets the Pattern Generator bit 0 output to toggle state.</p> <p><code>PG:BIT:ZERO?</code> might return <code>:PG:BIT:ZERO HIGH</code>, indicating the Pattern Generator bit 0 output is set to High.</p>

PG:BITRate

This command sets or queries the bit rate of data in the Pattern Generator. You can give any value, but it will take nearest possible value and generate the pattern.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	<code>PG:BITRate <NR3></code> <code>PG:BITRate?</code>
Arguments	<code><NR3></code> sets the Pattern Generator bit rate. The minimum bit rate is 1 b/s and maximum bit rate is 25 Mb/s. The default value is 125000 b/s.
Examples	<p><code>PG:BITRate 1000</code> sets the Pattern Generator bit rate to 1000 b/s.</p> <p><code>PG:BITRate?</code> might return <code>:PG:BITRate 1</code>, indicating the Pattern Generator bit rate is 1 b/s.</p>

PG:BURSt:CCOUnt

This command sets or queries the cycle count for Pattern Generator burst mode. This is only available when output is Burst.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	<code>PG:BURSt:CCOUnt <NR1></code> <code>PG:BURSt:CCOUnt?</code>
Arguments	<NR1> sets the cycle count for Pattern Generator burst mode. The minimum cycle value can be set is 1 and maximum cycles are 2000. The default value is 1.
Examples	<code>PG:BURSt:CCOUnt 1</code> sets the cycle count for Pattern Generator burst mode to 1. <code>PG:BURSt:CCOUnt?</code> might return <code>:PG:BURSt:CCOUnt 1</code> , indicating the cycle count for Pattern Generator burst mode is 1.

PG:BURSt:TRIGger (No Query Form)

This command manually starts the burst pattern. This is only available when output is Burst.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	<code>PG:BURSt:TRIGger</code>
Related Commands	PG:BURSt:CCOUnt
Examples	<code>PG:BURSt:TRIGger</code> triggers a burst pattern.

PG:FILE:PATtern

This command sets or queries the path of your data file to generate a digital pattern in file mode. Only .csv files are supported.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator

Syntax	<code>PG:FILE:PATTERN <QString></code> <code>PG:FILE:PATTERN?</code>
Arguments	<p><code><QString></code> is the file path that specifies the location of the specified instrument file.</p> <p>If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:</p> <ul style="list-style-type: none"> ■ Complete path specification. If the file argument begins with a drive designator (such as C), then the file name is interpreted as a full path. ■ Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
Examples	<p><code>PG:FILE:PATTERN "E:/2Power.csv"</code> specifies the path to the pattern file.</p> <p><code>PG:FILE:PATTERN?</code> might return <code>PG:FILE:PATTERN "E:/2Power.csv"</code>, indicating pattern file path.</p>

PG:OUTPut:MODE

This command sets or queries the Pattern Generator output mode.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	<code>PG:OUTPut:MODE {CONTinuous BURSt OFF}</code> <code>PG:OUTPut:MODE?</code>
Arguments	<p><code>CONTinuous</code> sets the Pattern Generator output type to continuous.</p> <p><code>BURSt</code> sets the Pattern Generator output type to burst.</p> <p><code>OFF</code> sets the Pattern Generator output type to off. This is the default value.</p>
Examples	<p><code>PG:OUTPut:MODE BURSt</code> sets the Pattern Generator mode to burst.</p> <p><code>PG:OUTPut:MODE?</code> might return <code>PG:OUTPut:MODE CONTinuous</code>, indicating the Pattern Generator mode is set to continuous.</p>

PG:PATTERNdefinition

This command sets or queries the Pattern Generator definition. If set to manual mode, you can configure the bit independently. If set to file mode, you can set a data file that will be loaded into memory and generated as a sequence.

Conditions	Requires option 2-SOURCE.
Group	Pattern Generator
Syntax	PG:PATTERNdefinition {MANua1 FILE} PG:PATTERNdefinition?
Arguments	MANua1 sets the pattern definition source to Manual. This is the default value. FILE sets the pattern definition source to File.
Examples	PG:PATTERNdefinition FILE sets the pattern definition to file mode. PG:PATTERNdefinition? might return PG:PATTERNdefinition MANua1, indicating the pattern definition source is set to manual mode.

PLOT:ADDNew (No Query Form)

This command adds the specified plot.

Group	Plot
Syntax	PLOT:ADDNew <QString>
Arguments	<QString> is the specified plot. The argument is of the form "PLOT<NR1>", where <NR1> \geq 1.
Examples	PLOT:ADDNEW "PLOT1" adds PLOT1.

PLOT:DELeTe (No Query Form)

This command deletes the specified plot.

Group	Plot
Syntax	PLOT:DELEte <QString>
Arguments	<QString> is the specified plot. Argument is of the form "PLOT<NR1>, where <NR1> is ≥ 1).
Examples	PLOT:DELETE "PLOT1" deletes PLOT1.

PLOT:LIST? (Query Only)

This command lists all currently defined plots.

Group	Plot
Syntax	PLOT:LIST?
Returns	A list of all currently defined plots is returned.
Examples	PLOT:LIST? might return :PLOT:LIST PLOT1, PLOT3, PLOT4, PLOT5, PLOT6, PLOT7 listing all currently defined plots.

PLOT:PLOT<x>:SOUrce<x>

This command sets or queries the plot source.

Group	Plot
Syntax	PLOT:PLOT<x>:SOUrce<x> MEAS<x> PLOT:PLOT<x>:SOUrce<x>?
Arguments	MEAS<x> is the specified measurement source for the specified plot.
Examples	PLOT:PLOT1:SOUrce1 MEAS2 sets source 1 of plot 1 to measurement 2. PLOT:PLOT1:SOUrce1? might return :PLOT:PLOT1:SOUrce1 MEAS1 indicating the specified source of the specified plot is measurement 1.

PLOT:PLOT<x>:TYPE

This command sets or returns the current plot type of the specified plot.

Group Plot

Syntax PLOT:PLOT<x>:TYPE {NONE|XY}

Arguments <x> is the plot number. This is the equivalent of the number shown on a plot heading in the UI.

NONE does not create a plot.

XY creates a XY plot.

Examples PLOT:PLOT2:TYPE XY creates a XY plot.

PLOT:PLOT2:TYPE? might return :PLOT:PLOT2:TYPE XY indicating the plot is an XY plot.

*PSC

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

Group Status and Error

Syntax *PSC {<NR1>|OFF|ON}
*PSC?

Related Commands [DESE](#)
[*ESE](#)
[FACTory](#)
[*RST](#)
[*SRE](#)

Arguments	<p><NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.</p> <p>OFF sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on.</p> <p>ON sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.</p>
Examples	<p>*PSC 0 sets the power-on status clear flag to false.</p> <p>*PSC? might return 1 to indicate that the power-on status clear flag is set to true.</p>

*PUD

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

Group Status and Error

Syntax *PUD {<Block>|<QString>}
*PUD?

Related Commands [PASSWord](#)

Arguments <Block> is a block containing up to 100 characters.
<QString> is a string containing up to 100 characters.

Examples *PUD #229This instrument belongs to me stores the string "This instrument belongs to me" in the user protected data area.
*PUD? might return #221PROPERTY OF COMPANY X.

RECALL:MASK (No Query Form)

This command recalls a saved mask definition from a Mask File. File suffixes can be xml or msk. If the specified mask test already exists the mask associated with that mask test will be replaced, otherwise a new mask test is created.

Group	Mask
Syntax	RECALL:MASK <source file>,MASK<x>
Arguments	<p>MASK<x> is the destination mask.</p> <p><source file> is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:</p> <ul style="list-style-type: none">■ Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.■ Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
Examples	RECALL:MASK "MaskWD17",MASK1 recalls the mask stored in the file named MASKWD17 and stores it in mask 1. If mask 1 already exists then its mask will be replaced.

RECALL:SESSion (No Query Form)

Restores the state of the instrument, including reference waveforms, from a saved session file.

Group	Save and Recall
Syntax	RECALL:SESSion <file_path>
Arguments	<p><file_path> is the file path that specifies the location of the specified instrument session file.</p> <p>If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:</p>

- Complete path specification. If the file argument begins with a drive designator (such as C), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Returns Recalls the instrument session from the specified session file.

Examples `RECALL:SESSION "TEK00000.TSS"` recalls the setup from the file `TEK00000.TSS` in the current working directory.

RECALL:SETUp (No Query Form)

This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. This command performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

Group Save and Recall

Syntax `RECALL:SETUp {FACTory|<file_path>}`

Related Commands [FACTory](#)
[*RST](#)

Arguments `FACTory` restores the factory setup. Performs the same operation as the `:FACTory` command.

`<file_path>` specifies a location for an instrument setup file. `<file path>` is a quoted string that defines the file name and path. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

- Examples** `RECALL:SETUP FACTORY` recalls (and makes current) the instrument setup to its factory defaults.
- `RECALL:SETUP "TEK00000.SET"` recalls the setup from the file `TEK00000.SET` in the default directory for setups.

RECALL:WAVEform (No Query Form)

This command recalls a stored waveform to a reference memory location.

Group Save and Recall

Syntax `RECALL:WAVEform <source file>,<destination>`

Arguments `<source file>` is the source file. The file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

`<destination>` is `REF<x>` which specifies a reference to create from the recalled waveform data file.

- Examples** `RECALL:WAVEFORM "TEK00000.ISF",REF1` recalls the waveform stored in the file named `TEK00000.ISF` from the current directory to reference 1.

REF:ADDNew (No Query Form)

This command adds the specified reference. Argument is of the form "`REF<NR1>`", where $NR1 \geq 1$.

Group Vertical

Syntax `REF:ADDNew <QString>`

Arguments <QString> is the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Examples REF:ADDNEW "REF2" adds reference 2 to the display.

REF:DELEte (No Query Form)

Deletes the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Conditions Vertical

Group Vertical

Syntax REF:DELEte <QString>

Arguments <QString> is the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Examples REF:DELETE "REF2" deletes reference 2 from the display.

REF:LIST? (Query Only)

This command returns a comma separated list of all currently defined references.

Group Vertical

Syntax REF:LIST?

Returns All currently defined references.

Examples REF:LIST? might return :REF:LIST REF1,REF2 indicating references 1 and 2 are defined.

REF:REF<x>:DESKew

This command sets or queries the deskew value used for the specified reference.

Group	Vertical
Syntax	REF:REF<x>:DESKew <NR3>
Arguments	<NR3> is the deskew value used for the specified reference.
Examples	REF:REF1:DESKew -1.5e-9 sets the deskew value to -1.5 ns. REF:REF1:DESKew? might return :REF:REF1:DESKew 1.5200E-9 indicating the deskew value is 1.52 ns.

REF:REF<x>:LABel:COLor

This command sets or queries the color of the specified ref label.

Group	Vertical
Syntax	REF:REF<x>:LABel:COLor <QString>
Arguments	<QString> is the label. To return the color to the default color, send an empty string as in this example: :REF:REF1:LABel:COLor "".
Examples	REF:REF1:LABel:COLor "#FFFF00" sets the font color to yellow. REF:REF1:LABel:COLor? might return :REF:REF1:LABel:COLor "#FF0000" indicating the font color is red.

REF:REF<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified reference label.

Group	Vertical
Syntax	REF:REF<x>:LABel:FONT:BOLD {<NR1> OFF ON}
Arguments	<NR1> = 0 disables bold font; any other value turns this feature on. OFF disables bold font.

ON enables bold font.

Examples REF:REF1:LABE1:FONT:BOLD ON turns on the bold font.

REF:REF1:LABE1:FONT:BOLD? might return :REF:REF1:LABEL:FONT:BOLD
0 indicating the bold font is off.

REF:REF<x>:LABE1:FONT:ITALIC

This command sets or queries the italic state of the specified reference label.

Group Vertical

Syntax REF:REF<x>:LABE1:FONT:ITALIC {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables italic font; any other value turns this feature on.

OFF disables italic font.

ON enables italic font.

Examples REF:REF1:LABE1:FONT:ITALIC ON turns on the italic font.

REF:REF1:LABE1:FONT:ITALIC? might return
:REF:REF1:LABEL:FONT:ITALIC 0 indicating the italic font is off.

REF:REF<x>:LABE1:FONT:SIZE

This command sets or queries the font size of the specified reference label.

Group Vertical

Syntax REF:REF<x>:LABE1:FONT:SIZE <NR1>

Arguments <NR1> is the font size of the label.

Examples REF:REF1:LABE1:FONT:SIZE 20 sets the font size to 20 points.

REF:REF1:LABE1:FONT:SIZE? might return :REF:REF1:LABEL:FONT:SIZE
14 indicating that the font size is 14 points.

REF:REF<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified reference label, such as Arial or Times New Roman.

Group Vertical

Syntax REF:REF<x>:LABel:FONT:TYPE <QString>

Arguments <QString> is the font type.

Examples REF:REF1:LABel:FONT:TYPE "Monospace" specifies a mono spaced font.
REF:REF1:LABel:FONT:TYPE? might return :REF:REF1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

REF:REF<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified reference label.

Group Vertical

Syntax REF:REF<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}

Arguments <NR1> = 0 disables underline font; any other value turns this feature on.
OFF disables underline font.
ON enables underline font.

Examples REF:REF1:LABel:FONT:UNDERline ON turns on the underline font.
REF:REF1:LABel:FONT:UNDERline? might return :REF:REF1:LABEL:FONT:UNDERLINE 0 indicating that underline is off.

REF:REF<x>:LABel:NAME

This command sets or queries the label of the specified reference. The reference waveform is specified by x.

Group	Vertical
Syntax	REF:REF<x>:LABel:NAME <QString> REF:REF<x>:LABel:NAME?
Arguments	<QString> is the character string that will be used for the reference waveform label name.
Examples	REF:REF4:LABEL:NAME "My Reference" sets the label name of Reference 4 waveform to "My Reference". REF:REF3:LABEL:NAME? might return :REF:REF3:LABEL:NAME "Signal2", indicating that the label name for Reference 3 waveform is currently set to "Signal2".

REF:REF<x>:LABel:XPOS

This command sets or queries the X-position at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the waveview. The reference waveform is specified by x.

Group	Vertical
Syntax	REF:REF<x>:LABel:XPOS <NR1> REF:REF<x>:LABel:XPOS?
Arguments	<NR1> is the location (control in divisions) where the waveform label for the selected reference is displayed, relative to the left edge of the screen.
Examples	REF:REF4:LABEL:XPOS 10 moves the waveform label for the Reference 3 waveform, so that it begins 10 divisions to the right of the left edge of the screen. REF:REF2:LABEL:XPOS? might return :REF:REF2:LABEL:XPOS 1.5, indicating that the x-axis for the Reference 2 waveform is currently 1.5 divisions to the right of the left edge of the screen.

REF:REF<x>:LABel:YPOS

This command sets or queries the Y-position of the label (attached to the displayed waveform of the specified reference), relative to the baseline of the waveform. The reference waveform is specified by x.

Group	Vertical
Syntax	REF:REF<x>:LABe1:YPOS <NR1> REF:REF<x>:LABe1:YPOS?
Arguments	<NR1> is the location where the waveform label for the selected reference is displayed, relative to the baseline of the waveform.
Examples	REF:REF3:LABEL:YPOS -10 moves the waveform label for the Reference 3 waveform 10 vertical units below the baseline of the waveform. REF:REF2:LABEL:YPOS? might return :REF:REF2:LABEL:YPOS 0, indicating that the waveform label for the Reference 2 waveform is currently located at the baseline of the waveform.

REF:REF<x>:SOUrce

This command sets or queries the filename used by the given reference.

Group	Vertical
Syntax	REF:REF<x>:SOUrce <QString>
Arguments	<QString> is the reference file name.
Examples	REF:REF1:SOUrce "/home/guest/.local/share/Tektronix/TekScope/ FirstRecalledSession/161012_132000_000.wfm" sets the source of the reference. REF:REF1:SOUrce? might return :REF:REF1:SOUrce "/home/guest/.local/share/Tektronix/TekScope/ LastRecalledSession/161012_132039_000.wfm".

REM (No Query Form)

This command (no query form) embeds a comment within programs as a means of internally documenting the programs. This is how to embed comments in a .set file. The instrument ignores these embedded comment lines.

Group	Miscellaneous
Syntax	REM <QString>
Arguments	<QString> is a string that can contain a maximum of 80 characters.
Examples	REM "This is a comment" is a comment string that the instrument will ignore.

ROSc:SOUrce

This command sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source. Depending on the command argument that you specify, you can use an external reference or use the internal crystal oscillator as the time base reference.

Group	Miscellaneous
Syntax	ROSc:SOUrce {INTERNAL EXTERNAL} ROSc:SOUrce?
Related Commands	ROSc:STATE?
Arguments	INTERNAL specifies the internal 10 MHz crystal oscillator as the time base reference. EXTERNAL specifies the user-supplied external signal as the time base reference.
Examples	ROSc:SOUrce INTERNAL specifies the internal 10 MHz crystal oscillator as the time base reference. ROSc:SOUrce? might return :ROSc:SOUrce INTERNAL, indicating that the 10 MHz crystal oscillator is being used as the time base reference.

ROSc:STATE? (Query Only)

This query-only command returns whether the time base reference oscillator is locked. This command will return either LOCKED or UNLOCKED.

Group	Miscellaneous
--------------	---------------

Syntax	ROSC:STATE?
Returns	LOCKED indicates the reference oscillator is locked. UNLOCKED indicates the reference oscillator is not locked.
Examples	ROSC:STATE? might return :ROSC:STATE LOCKED, indicating that the time base reference is locked.

*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings.

This command does the following:

- Recalls the default instrument setup.
- Clears the current *DDT command.
- Disables aliases (:ALIAS:STATE 0).
- Disables the user password (for the *PUD command).

The *RST command does not change the following:

- The current working directory (:FILESystem:CWD command).
- The state of command headers (:HEADer command).
- The state of keyword and enumeration verbosity (:VERBOse command).
- The Power-on Status Clear Flag (*PSC command).
- The Event Status Enable Register (*ESE command).
- The Service Request Enable Register (*SRE command).
- The Device Event Status Enable Register (DESE command).
- The user password for protected user data (:PASSWOrd command).
- The content of protected user data (*PUD command).
- The enabled state of the socket server (:SOCKETServer:ENAbLe command).
- The socket server port number (:SOCKETServer:PORT command).
- The socket server protocol (:SOCKETServer:PROTOCOL command).
- The USBTMC port configuration (:USBDeVice:CONFIgure command).
- The destination reference waveform or file path for the :CURVe command (:DATA:DESTInation command).
- The source waveform for the :CURVe? or :WAVFrm? queries (:DATA:SOURce command).
- The waveform data encoding for the :CURVe command or query or the :WAVFrm? query (:DATA:ENCdg command).
- The starting point for :CURVe? queries (:DATA:STARt command).
- The ending point for :CURVe? queries (:DATA:STOP command).
- All settings associated the :WFMIpre commands.
- All user settable settings associated with the WFMOutpre commands.

*RST only resets the programmable interface settings, it does not change the user interface settings.

Group Status and Error

Syntax *RST

Related Commands [FACtory](#)
[RECAIl:SETUp](#)

Arguments None

Examples *RST resets the instrument settings to factory defaults.

SAVE:EVENTtable:BUS (No Query Form)

This command saves bus results table to the specified file.

Group Save and Recall

Syntax SAVE:EVENTtable:BUS <QString>

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples SAVE:EVENTtable:BUS "TEK000.CSV" saves the bus decode event table in the file named TEK000.CSV.

SAVE:EVENTtable:MEASUREMENT (No Query Form)

This command saves data (measurement) results to the specified file.

Group Save and Recall

Syntax SAVE:EVENTtable:MEASUREMENT <QString>

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVE:EVENTtable:MEASUREMENT "TEK000.CSV"` saves the measurement in the file named TEK000.CSV.

SAVE:EVENTtable:SEARCHTable (No Query Form)

This command saves a search results table to the specified file.

Group Save and Recall

Syntax `SAVE:EVENTtable:SEARCHTable <QString>`

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVE:EVENTtable:SEARCHTable "C:\Users\Tek_Local_Admin\Tektronix\TekScope\Results Tables\Tek002.csv"` saves the search results table in the file named TEK002.CSV.

SAVE:IMAGe (No Query Form)

Saves a capture of the screen contents to the specified image file. Supported image formats are PNG, Windows Bitmap, and JPEG.

Group Save and Recall

Syntax `SAVE:IMAGE <QString>`

Arguments `<QString>` is the file name and location used to store the image file.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by `:FILESystem:CWD`[FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVE:IMAGE "C:/Dut12-tests.png"` saves the image at the location specified.

SAVE:IMAGe:COMPosition

Sets or queries the color mode for saved images (normal or inverted).

Group Save and Recall

Syntax `SAVE:IMAGe:COMPosition {NORMa|INVERTed}`
`SAVE:IMAGe:COMPosition?`

Arguments `NORMa|` Sets the saved screen capture to Normal colors.

`INVERTed` sets the saved screen capture to Inverted colors.

Examples `SAVE:IMAGe:COMPOSITION NORMa|` saves the screen image using the current normal color settings.

`SAVE:IMAGe:COMPOSITION?` might return `INVERTED`, indicating that captured screen images will be saved using the inverted color settings.

SAVE:IMAGe:VIEWTYpe

Sets or queries the view type for saved images. Currently only `FULLScreen` is supported.

Group	Save and Recall
Syntax	SAVE:IMAGe:VIEWType {FULLScreen} SAVE:IMAGe:VIEWType?
Arguments	FULLScreen sets the screen capture mode to capture the full screen.
Examples	SAVE:IMAGe:VIEWType FULLScreen sets the screen capture mode to capture the full screen. SAVE:IMAGe:VIEWType? might return FULLScreen, indicating that the screen capture mode is set to full screen.

SAVe:PLOTData (No Query Form)

Saves the plot data of the currently selected plot to a specified file. Supported file format is CSV.

Group	Save and Recall
Syntax	SAVe:PLOTData <Qstring>
Related Commands	DISplay:SElect:VIEW FILESystem:CWD
Arguments	<p>Qstring sets the file name and location used to store the plot data. When specifying the file name with this command, use the correct file extension (.CSV). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.</p> <p>If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path. If the file argument begins with "." or "", or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.</p> <p>To export an eye diagram plot data to a .csv file, the prerequisite command is MEASurement:ADDMEAS TIE</p>
Examples	SAVe:PLOTDATA "plot1.csv" saves the plot to the designated file in the current working directory.

```
The following is the example of exporting an XY plot data to a .csv file:  
PLOT:PLOT1:TYPE XY  
DISplay:SElect:VIEW PLOTVIEW1  
SAVe:PLOTData "C:/plot1.csv"
```

SAVE:REPOrt (No Query Form)

This command saves a report to the specified file. Supported report formats are PDF and MHT (web page archive file).

Group Save and Recall

Syntax SAVE:REPOrt <QString>

Related Commands [SAVE:REPOrt:COMMENTS](#)

Arguments <QString> is the complete path specification. When specifying the file name with this command, use the correct file extension (.pdf for PDF format, or .mht for MHT format).

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples SAVE:REPORT "report.pdf" creates a report in PDF format, in the location specified.

SAVE:REPOrt:COMMENTS

This command sets or queries the comments to be included in saved report files.

Group Save and Recall

Syntax	<code>SAVE:REPORT:COMMENTS <QString></code> <code>SAVE:REPORT:COMMENTS?</code>
Arguments	<code><QString></code> is the comments to be included in saved report files.
Examples	<code>SAVE:REPORT:COMMENTS "Test 3"</code> adds comments to the report. <code>SAVE:REPORT:COMMENTS?</code> might return <code>:SAVE:REPORT:COMMENTS "High Temp Test 1"</code> .

SAVE:SESSion (No Query Form)

Saves the state of the instrument, including reference waveforms, to a saved session file.

Group	Save and Recall
Syntax	<code>SAVE:SESSion <QString></code>
Arguments	<code><QString></code> is the file path that specifies the location to save the specified instrument session file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by <code>FILESystem:CWD</code> FILESystem:CWD) unless a complete path is specified: <ul style="list-style-type: none">■ Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.■ Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
Examples	<code>SAVE:SESSion "c:/rose_was_here.tss"</code> saves the instrument state in the specified file.

SAVE:SETUp (No Query Form)

Saves the current instrument state to the specified file.

Group	Save and Recall
--------------	-----------------

Syntax SAVE:SETUP <QString>

Related Commands [SAVE:SETUp:INCLUDEREFs](#)

Arguments <QString> is a quoted string that is the complete path specification. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples SAVE:SETUP "c:/rose_was_here.set" saves the instrument setup in the specified file.

SAVE:SETUp:INCLUDEREFs

This command sets or queries whether displayed reference waveforms are to be included in saved setups.

Group Save and Recall

Syntax SAVE:SETUp:INCLUDEREFs {OFF|ON|0|1}
SAVE:SETUp:INCLUDEREFs?

Arguments OFF specifies not including displayed reference waveforms in saved setups.
ON specifies including displayed reference waveforms in saved setups.
0 specifies not including displayed reference waveforms in saved setups.
1 specifies including displayed reference waveforms in saved setups.

Examples SAVE:SETUp:INCLUDEREFs 0 sets reference waveforms not to be included in saved setups.

SAVE:SETUp:INCLUDEREFs? might return :SAVE:SETUp:INCLUDEREFs 1 indicating that reference waveforms are to be included in saved setups.

SAVe:WAVEform (No Query Form)

This command saves the specified waveform(s) to the specified destination file. The waveform source or sources must be active (turned on) to save data to a file.

Conditions `_MAG_VS_TIME`, `_FREQ_VS_TIME`, `_PHASE_VS_TIME`, and `_SV_BASEBAND_IQ` require option SV-RFVT.

Option SV-RFVT on the 4 Series MSO instruments requires option SV-BAS.

Group Save and Recall

Syntax `SAVe:WAVEform`
`{CH<x>[_DALL|_SV_NORMa1|_SV_AVErAge|_SV_MAXHo1d|`
`_SV_MINHo1d|_MAG_VS_TIME|_FREQ_VS_TIME|`
`_PHASE_VS_TIME]|MATH<x>|REF<x>|ALL| },<QString>`

Related Commands [FILESystem:CWD](#)

[SAVe:WAVEform:SOURCEList?](#)

Arguments `<x>` is the number of the analog channel, math, or reference waveform source used to save the waveform data.

`_DALL` saves the digital channel waveform data of the specified channel. This argument is required if the channel specified is a digital channel.

`_SV_NORMa1` saves the Normal Spectrum view waveform of the specified channel.

`_SV_AVErAge` saves the Average Spectrum view waveform of the specified channel.

`_SV_MAXHo1d` saves the Maximum Hold Spectrum view waveform of the specified channel.

`_SV_MINHo1d` saves the Minimum Hold Spectrum view waveform of the specified channel.

`_MAG_VS_TIME` saves the Magnitude vs. Time waveform of the specified channel.

`_FREQ_VS_TIME` saves the Frequency vs. Time waveform of the specified channel.

`_PHASE_VS_TIME` saves the Phase vs. Time waveform of the specified channel.

`_SV_BASEBAND_IQ` saves the baseband I & Q data of the specified channel. The data is saved in Tektronix TIQ format using a .TIQ file extension. You can import .TIQ files into Tektronix SignalVu-PC software (PC based), SignalVu software (oscilloscope based), RSAVu software (PC based) or into a Tektronix real-time spectrum analyzer for pulse analysis and demodulation analysis.

ALL saves all displayed analog, math, and reference waveforms to individual files. Each file name created includes the name of the source (ch1, math3, and so on) used to create that file. ALL is not supported when FastAcq Mode is enabled.

<Qstring> is a quoted string that defines the path and file name to use to save the specified file, in the format '[<path>]<filename.ext>'. Specifying a path is optional. If no path is entered, the file is saved to the current working directory as set in [FILESystem:CWD](#).

<path> uses the form '<drive>/<dir>.../'. You can specify a relative path or a complete path:

- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters. When using the ALL argument to save multiple files, each filename has the filename appended with the source used to create that file. For example, a filename of QualTest can create QualTest_ch1.xxx, QualTest_ref1.xxx, and so on.

<.ext> sets the file format to which to save the data. To save Channel FastAcq data, only .csv format is supported. The saved fastacq pixmap data .csv file cannot be recalled.

- Use the .wfm extension to save waveform data to a Tektronix Internal format.
- Use the .csv extension to save waveform data to a comma separated values spreadsheet format.
- Use the .mat extension to save waveform data to a matlab compatible file format.

Examples SAVE:WAVEFORM MATH1, "TEK0000.WFM" saves the Math1 waveform to the file TEK0000.WFM in the current working directory.

SAVE:WAVEform:GATing

This command specifies the method to save a specified part of the waveform data or the entire waveform.

Group Save and Recall

Syntax	<code>SAVE:WAVEform:GATing {NONE CURSors SCREEN RESAMPLE SElected}</code> <code>SAVE:WAVEform:GATing?</code>
Arguments	<p><code>NONE</code> saves the full waveform data.</p> <p><code>CURSors</code> saves the waveform data located between the vertical cursors.</p> <p><code>SCREEN</code> saves the waveform data that is on the screen. Nothing outside the waveform will be saved.</p> <p><code>RESAMPLE</code> saves the waveform data at a sample interval set by the user. The resulting saved waveform is a resampled version of the original waveform with fewer data points.</p> <p><code>SElected</code> saves the data from the currently selected history or FastFrame acquisition.</p>
Examples	<p><code>SAVE:WAVEFORM:GATING NONE</code> saves the entire waveform.</p> <p><code>SAVE:WAVEFORM:GATING?</code> might return <code>:SAVE:WAVEform:GATing SCREEN</code>, indicating the save waveform operation is set to save the waveform data on the screen.</p>

SAVE:WAVEform:GATing:RESAMPLERate

This command saves the waveform data at a sample interval. The resulting saved waveform is a resampled version of the original waveform with fewer data points.

Group	Save and Recall
Syntax	<code>SAVE:WAVEform:GATing:RESAMPLERate <NR1></code> <code>SAVE:WAVEform:GATing:RESAMPLERate?</code>
Arguments	<NR1> specifies the resample interval.
Examples	<p><code>SAVE:WAVEFORM:GATING:RESAMPLERATE 3</code> sets to save waveform data every third data point.</p> <p><code>SAVE:WAVEFORM:GATING:RESAMPLERATE?</code> might return <code>:SAVE:WAVEform:GATing:RESAMPLERate 2</code>, indicating the saved waveform will have half as many data points. This command saves the first data point and then every other following data point.</p>

SAVE:WAVEform:SOURCEList? (Query Only)

This query returns a list of the available waveforms that can be specified as the source for the SAVE:WAVEform command. Source waveforms must have their display mode set to On to appear in this list and to be saved.

Group	Save and Recall
Syntax	SAVE:WAVEform:SOURCEList?
Examples	SAVE:WAVEFORM:SOURCELIST? might return ALL, CH1, CH2, CH3, MATH1, MATH2, REF2, REF4, DCH1_DALL ..

SAVEON:FILE:DEST

This command sets or queries the location where files are saved when SAVEON:TRIGGER is ON and SAVEON:WAVEFORM is ON. You can save the files to a local drive or network path by entering the desired location in <QString>. You can also select to save the files to a USB drive.

Group	Save On
Syntax	SAVEON:FILE:DEST <QString> SAVEON:FILE:DEST?
Related Commands	SAVEON:FILE:NAME
Arguments	<QString> specifies the location to store files.
Examples	SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEON:FILE:NAME command), when there is a trigger. SAVEON:FILE:DEST? might return :SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnEvent", indicating the drive location where files will be saved when there is a trigger.

SAVEON:FILE:NAME

Sets or queries the file name to use when SAVEON:TRIGger is ON.

Group Save On

Syntax SAVEON:FILE:NAME <QString>
SAVEON:FILE:NAME?

Related Commands [SAVEON:FILE:DEST](#)

Arguments <QString> is the file name you want to use.

Examples SAVEON:FILE:NAME "MaskFailure" sets the name of the file to MaskFailure.
SAVEON:FILE:NAME? might return ":SAVEON:FILE:NAME MaskFailure5", indicating the name you set for the instrument to use, with the autoincrement number (5) appended.

SAVEON:IMAGe

This command sets or queries whether to save a screen capture when a trigger occurs and SAVEON:TRIGger is ON and SAVEON:IMAGe is ON.

Group Save On

Syntax SAVEON:IMAGe {<NR1>|OFF|ON}
SAVEON:IMAGe?

Related Commands [SAVEON:FILE:DEST](#)
[SAVEON:FILE:NAME](#)
[SAVEON:TRIGger](#)
[SAVEON:IMAGe](#)

Arguments <NR1> = 0 disables Save On Image; any other value turns this feature on.
OFF disables Save On Image.
ON enables Save On Image.

- Examples** `SAVEON:IMAGE ON` sets the instrument to save a screen capture on a specified trigger.
- `SAVEON:IMAGE?` might return `:SAVEON:IMAGE 1`, indicating that the instrument will save a screen capture when the specified trigger occurs.

SAVEON:IMAGe:FILEFormat

This command sets or queries the file format to be used for saved image files when `:SAVEON:IMAGe` is set to 1.

- Group** Save On
- Syntax** `SAVEON:IMAGe:FILEFormat {PNG|BMP|JPG}`
`SAVEON:IMAGe:FILEFormat?`
- Arguments** PNG specifies using PNG format for saved image files.
BMP specifies using BMP format for saved image files.
JPG specifies using JPEG format for saved image files.
- When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by [FILESystem:CWD](#)) unless a complete path is specified:
- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
 - Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
- Examples** `SAVEON:IMAGe:FILEFORMAT JPG` sets the image file format to JPEG.
- `SAVEON:IMAGe:FILEFORMAT?` might return `:SAVEON:IMAGe:FILEFORMAT PNG` indicating that the file format is set to PNG.

SAVEON:TRIGger

Sets or queries whether to save a file when a trigger occurs. You can define the trigger using Trigger commands or the instrument user interface.

This command is longer necessary. Please see Act On Event commands for future development.

The trigger will cause the instrument to save an image or a waveform to a file, depending on what you specified. For example, if you have set [SAVEON:IMAGe](#) to On, and a trigger event occurs, the instrument will save a screen capture. You can set options for file storage (such as file name, file destination, and auto increment), using the [SAVEON:FILE](#) commands.

Use the instrument interface to select whether to save one or more analog channels, digital channels, or math waveforms

Analog and math waveforms are saved using one file per waveform. Digital waveforms are all saved to a single file.

Group Save On

Syntax `SAVEON:TRIGger {<NR1>|ON|OFF}`
`SAVEON:TRIGger?`

Related Commands [SAVEON:IMAGe](#)
[SAVEON:WAVEform](#)
[SAVEON:FILE:DEST](#)
[SAVEON:FILE:NAME](#)

Arguments <NR1> = 0 disables Save On Trigger; any other value turns this feature on.
OFF disables Save On Trigger.
ON enables Save On Trigger.

Examples `SAVEON:TRIGGER ON` sets the instrument to save an image, measurement, and/or waveform when a trigger occurs.
`SAVEON:TRIGGER?` might return `:SAVEON:TRIGGER ON`, indicating that a file will be saved on triggering.

SAVEON:WAVEform

Sets or queries whether to save a waveform when a trigger occurs when [SAVEON:TRIGger](#) is ON.

The waveform will be saved to the file you selected with [SAVEON:FILE:NAME](#), in the location that you selected using [SAVEON:FILE:DEST](#). You can set options

for file storage (such as file name, file destination, and autoincrement), using the `SAVEON:FILE` commands.

Group	Save On
Syntax	<code>SAVEON:WAVEform {<NR1> ON OFF}</code> <code>SAVEON:WAVEform?</code>
Related Commands	SAVEON:FILE:DEST SAVEON:FILE:NAME SAVEON:TRIGger
Arguments	<code><NR1> = 0</code> disables Save On Waveform; any other value turns this feature on. <code>OFF</code> disables Save On Waveform. <code>ON</code> enables Save On Waveform.
Examples	<code>SAVEON:WAVEFORM ON</code> turns on the Save On Waveform feature, so that a waveform will be saved when a selected trigger occurs. <code>SAVEON:WAVEFORM?</code> might return <code>:SAVEON:WAVEFORM ON</code> , indicating that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEform:FILEFormat

This command sets or queries the file format for saving waveforms when `:SAVEON:WAVEform` is set to `1`.

Group	Save On
Syntax	<code>SAVEON:WAVEform:FILEFormat {INTERNAL SPREADSheet}</code> <code>SAVEON:WAVEform:FILEFormat?</code>
Arguments	<code>INTERNAL</code> specifies saving the waveform in the instrument internal format. <code>SPREADSheet</code> specifies saving the waveform in comma separated values format.
Examples	<code>SAVEON:WAVEFORM:FILEFORMAT SPREADSheet</code> sets the file format to spreadsheet.

SAVEON:WAVEFORM:FILEFORMAT? might return
:SAVEON:WAVEFORM:FILEFORMAT INTERNAL indicating the file format is set
to INTERNAL.

SAVEON:WAVEform:SOURce

This command sets or queries the sources for saving waveforms when [SAVEON:TRIGger](#) is ON.

Group	Save On
Syntax	SAVEON:WAVEform:SOURce {CH<x> DCH<x>_D<x> MATH<x> REF<x> ALL} SAVEON:WAVEform:SOURce?
Arguments	<p>CH<x> specifies an analog channel to use as the save on source.</p> <p>DCH<x>_D<x> specifies a digital channel to use as the save on source. The supported digital channel value is 1. The supported digital bit values are 0 to 15 or DALL (all digital channels).</p> <p>MATH<x> specifies a math waveform to use as the save on source.</p> <p>REF<x> specifies a reference waveform to use as the save on source.</p> <p>ALL specifies all analog, math, digital, and reference waveforms as the source waveforms for saving.</p>
Examples	<p>SAVEON:WAVEform:SOURce MATH1 specifies MATH 1 as the save on source.</p> <p>SAVEON:WAVEform:SOURce? might return :SAVEON:WAVEFORM:SOURCE REF1 indicating the save on source is REF1.</p>

SAVEONEVent:FILEDest

This command sets or queries the location where files are saved. This command replaces [SAVEON:FILE:DEST](#) (still valid command, but only an alias for this new command).

Group	Act On Event
Syntax	SAVEONEVent:FILEDest SAVEONEVent:FILEDest? <Qstring>

Arguments	<QString> specifies the location to store files.
Examples	<p>SAVEONEVENT:FILEDEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEONEVENT:FILENAME command), when there is a trigger.</p> <p>SAVEONEVENT:FILEDEST? might return :SAVEONEVENT:FILEDEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger", indicating the drive location where files will be saved when there is a trigger.</p>

SAVEONEVENT:FILENAME

This command sets or queries the file name without the extension. This command replaces [SAVEON:FILE:NAME](#) (still valid command, but only an alias for this new command).

Group	Act On Event
Syntax	SAVEONEVENT:FILENAME <QString> SAVEONEVENT:FILENAME?
Arguments	<QString> specifies the name of the file.
Examples	<p>SAVEONEVENT:FILENAME "MaskFailure" sets the name of the file to MaskFailure.</p> <p>SAVEONEVENT:FILENAME? might return :SAVEONEVENT:FILENAME "MaskFailure5", indicating the name you set for the instrument to use. The autoincrement number of 5 is appended.</p>

SAVEONEVENT:IMAGE:FILEFORMAT

This command sets or returns the image file extension (png, jpg, bmp). This command replaces [SAVEON:IMAGE:FILEFORMAT](#) (still valid command, but only an alias for this new command).

Group	Act On Event
Syntax	SAVEONEVENT:IMAGE:FILEFORMAT {PNG BMP JPG} SAVEONEVENT:IMAGE:FILEFORMAT?

Arguments	<p>PNG specifies using PNG format for saved image files.</p> <p>BMP specifies using BMP format for saved image files.</p> <p>JPG specifies using JPEG format for saved image files.</p> <p>When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:</p> <ul style="list-style-type: none"> ■ Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path. ■ Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
Examples	<p>SAVEONEVENT:IMAGE:FILEFORMAT JPG sets the image file format to JPEG.</p> <p>SAVEONEVENT:IMAGE:FILEFORMAT? might return :SAVEON:IMAGE:FILEFORMAT PNG indicating that the file format is set to PNG.</p>

SAVEONEVENT:WAVEFORM:FILEFORMAT

This command sets or returns the file extension (csv, wfm, mat). This command replaces [SAVEON:WAVEFORM:FILEFORMAT](#) (still valid command, but only an alias for this new command).

Group	Act On Event
Syntax	<pre>SAVEONEVENT:WAVEFORM:FILEFORMAT {INTERNAL SPREADSHEET MATLAB} SAVEONEVENT:WAVEFORM:FILEFORMAT?</pre>
Arguments	<p>INTERNAL specifies saving the waveform in the instrument internal format.</p> <p>SPREADSHEET specifies saving the waveform in comma separated values format.</p> <p>MATLAB specifies saving the waveform in matlab compatible file format.</p>
Examples	<p>SAVEONEVENT:WAVEFORM:FILEFORMAT SPREADSHEET sets the file format to spreadsheet (csv).</p>

SAVEONEVENT:WAVEFORM:FILEFORMAT? might return
:SAVEONEVENT:WAVEFORM:FILEFORMAT INTERNAL, indicating the
file format is set to INTERNAL.

SAVEONEVENT:WAVEFORM:SOURCe

This command sets or returns the sources for saving waveforms when an event occurs. This command replaces [SAVEON:WAVEFORM:SOURCe](#) (still valid command, but only an alias for this new command).

Group	Act On Event
Syntax	SAVEONEVENT:WAVEFORM:SOURCe {CH<x> DCH<x>_D<x> MATH<x> REF<x> ALL} SAVEONEVENT:WAVEFORM:SOURCe?
Arguments	<p>CH<x> specifies an analog channel as the source waveform for saving.</p> <p>DCH<x>_D<x> specifies a digital channel to use as the save on source. The supported digital channel value is 1. The supported digital bit values are 0 to 15 or DALL (all digital channels).</p> <p>Math<x> specifies a math waveform as the source waveform for saving.</p> <p>REF<x> specifies a reference waveform as the source waveform for saving.</p> <p>ALL specifies all analog, math, digital, and reference waveforms as the source waveforms for saving.</p>
Examples	<p>SAVEONEVENT:WAVEFORM:SOURCe MATH1 specifies MATH 1 as the save on source.</p> <p>SAVEONEVENT:WAVEFORM:SOURCe? might return :SAVEONEVENT:IMAGE:FILEFORMAT REF1, indicating the save on source is REF 1.</p>

SCOPEApp REBOOT (No Query Form)

This command reboots the scope.

Group	Miscellaneous
Syntax	SCOPEApp REBOOT

Examples SCOPEApp REBOOT reboots the scope.

SEARCH:ADDNew (No Query Form)

This command adds the specified search.

Group Search and Mark

Syntax SEARCH:ADDNew <QString>

Arguments <QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1 .

Examples SEARCH:ADDNEW "SEARCH2" adds a new search named SEARCH 2.

SEARCH:DELEte (No Query Form)

This command deletes the specified search.

Group Search and Mark

Syntax SEARCH:DELEte <QString>

Arguments <QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1).

Examples SEARCH:DELETE "SEARCH3" deletes SEARCH 3.

SEARCH:DELETEALL (No Query Form)

This command deletes all the active instances of search definitions defined in the scope application.

Group Search and Mark

Syntax SEARCH:DELETEALL

Examples `SEARCH:DELETEALL` deletes all the active instances of search definitions.

SEARCH:LIST? (Query Only)

This command returns a comma separated list of all currently defined searches.

Group Search and Mark

Syntax `SEARCH:LIST?`

Returns All currently defined searches.

Examples `SEARCH:LIST?` might return `:SEARCH:LIST SEARCH1,SEARCH2` indicating that Search 1 and Search 2 are defined.

SEARCH:SEARCH<x>:COpy (No Query Form)

This command (no query form) copies the search criteria to or from the trigger. The search number is specified by x.

Group Search and Mark

Syntax `SEARCH:SEARCH<x>:COpy {SEARCHtotrigger|TRIGgertosearch}`

Arguments `SEARCHtotrigger` copies the search criteria to the trigger.
`TRIGgertosearch` copies the trigger criteria to the search.

Examples `SEARCH:SEARCH1:COpy TRIGGERTOSEARCH` copies the trigger criteria to the search 1 criteria.

`SEARCH:SEARCH1:COpy SEARCHTOTRIGGER` copies the search criteria to the trigger.

SEARCH:SEARCH<x>:NAVigate (No Query Form)

This command sets the navigation action for search marks. The `NONE` action is the default setting when no action is being taken. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:NAVigate {NEXT PREVIOUS MIN NONE MAX}
Arguments	<p>NEXT goes to the next search mark.</p> <p>PREVIOUS goes to the previous search mark.</p> <p>MIN goes to the search result with the smallest value. Only supported by search results which have quantitative values (example: pulse width is supported, but not edge).</p> <p>NONE is the default setting when no action is being taken.</p> <p>MAX goes to the search result with the largest value. Only supported by search results which have quantitative values (example: pulse width is supported, but not edge).</p>
Examples	SEARCH:SEARCH1:NAVigate NEXT goes to the next search mark.

SEARCH:SEARCH<x>:TOTAL? (Query Only)

This query-only command returns the total number of found search marks for this search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TOTAL?
Examples	SEARCH:SEARCH1:TOTAL? might return SEARCH:SEARCH1:TOTAL 7, indicating that there are 7 matches for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition

This command sets or queries the search condition for a CAN bus. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition
{SOF|FRAMetype|IDentifier|DATA|IDANDDATA|EOF|ERRor|FDBITS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition?

Arguments Arguments specify the CAN bus trigger condition.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition EOF sets the CAN bus trigger condition to end of frame.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:CONDITION SOF, indicating the CAN bus trigger condition is set to start of frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:DIRection

This command specifies the CAN search type to be valid on a Read, Write, or Either condition. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:DIRection
{READ|WRITE|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:DIRection?

Arguments READ specifies the read direction.

WRITE specifies the write direction.

NOCARE specifies either data direction.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:DIRection READ sets the data direction to READ.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:DIRection? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:DIRECTION NOCARE,
indicating the data direction is set to either data direction.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:OFFSet

This command sets or queries the data offset value, in bytes, to use when searching on the CAN data field. The search number is specified by x. The search condition must be set to DATA or IDANDDATA.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:OFFSet?

Related Commands [BUS:B<x>:CAN:STANDard](#)
[SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:SIZE](#)
[SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:VALue](#)

Arguments <NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD). The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSet 5sets the CAN data offset to 5 bytes.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSet? might return 7, indicating the CAN data offset is 7 bytes. If the CAN standard is set for CAN 2.0 and the search data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5). If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD and the search data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:QUALifier

This command sets or queries the CAN bus trigger data qualifier to be used when searching on a CAN bus signal. The search number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:QUALIFIER
{EQUAL|LESSEQUAL|MOREEQUAL|UNEQUAL|LESSTHAN|MORETHAN}
SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:QUALIFIER?

Arguments Arguments are the data qualifier types.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:QUALIFIER UNEQUAL sets the data qualifier to unequal.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:QUALIFIER? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:QUALIFIER EQUAL,
indicating that the data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:SIZE

This command sets or queries the length of the data string, in bytes, to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA or DATA. The search number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:SIZE <NR1>
SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:SIZE?

Arguments <NR1> specifies the data size.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:SIZE 1 sets the data size to 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:SIZE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:SIZE 1, indicating the data size is set to 1.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:CAN:DATA:VALUE

This command sets or queries the binary data value to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA OR DATA.

Group Search and Mark

Syntax	<code>SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATA:VALue <QString></code>
Arguments	<code><QString></code>
Examples	<p><code>SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:VALue "1111"</code> sets the data value to 1111.</p> <p><code>SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATA:VALue?</code> might return <code>SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:VALUE "1010"</code> indicating the data value is 1010.</p>

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	<code>SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType</code> <code>{ACKMISS BITSTUFFing FORMERror ANYERror}</code> <code>SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType?</code>
Arguments	<p>ACKMISS specifies a search based on a missing ACK field.</p> <p>BITSTUFFing specifies a search based on a bit stuffing error.</p> <p>FORMERror specifies a search based on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.</p> <p>ANYERror specifies a search based on any error type.</p>
Examples	<p><code>SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRType ACKMISS</code> specifies searching for any missing ACK fields.</p> <p><code>SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRType?</code> might return ANYERror, indicating that the bus is being searched for all error types.</p>

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit {ONE ZERO NOCARE} SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit?
Arguments	ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled). ZERO filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled). NOCARE disables filtering of CAN FD packets on the BRS bit.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1. SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the BRS bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit {ONE ZERO NOCARE}

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit?

- Arguments** ONE filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).
- ZERO filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).
- NOCARE disables filtering of CAN FD packets on the ESI bit.
- Examples** SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBit ONE specifies filtering CAN FD packets for those where the ESI bit has a value of 1.
- SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMetype

This command sets or queries CAN bus trigger frame type to be used when searching on a CAN bus signal. The search condition must be set to FRAMetype. The search number is specified by x.

- Conditions** Requires option 2-SERIAL or 2-ULTIMATE.
- Group** Search and Mark
- Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMetype
{DATA|ERROR|OVERLoad|REMOte}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMetype?
- Arguments** Arguments are the available frame types.
- Examples** SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMetype ERROR sets the frame type to error.
- SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMetype? might return SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FRAMETYPE DATA, indicates the frame type is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODE

This command sets or queries the CAN bus trigger identifier (address) mode to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODE {EXTENDED STANDARD} SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODE?
Arguments	EXTENDED specifies the extended identifier mode. STANDARD specifies the standard identifier mode.
Examples	SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODE EXTEND sets the identifier mode to extended. SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:MODE STANDARD, indicating the identifier mode is set to standard.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue

This command sets or queries CAN bus trigger identifier (address) value to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue?
Arguments	<QString> is the identifier value.

- Examples** `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:VALUE "1010"` sets the identifier value to 1010.
- `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:VALUE "101011"`, indicating the identifier value is 101011.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:ADDRESS:MODE

This command sets or queries the I2C address mode for the specified bus search to determine where to place a mark. The search number is specified by x.

- Conditions** Requires option 2-SERIAL or 2-ULTIMATE.
- Group** Search and Mark
- Syntax** `SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:ADDRESS:MODE`
 `{ADDR10|ADDR7}`
 `SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:ADDRESS:MODE?`
- Arguments** `ADDR10` specifies the address mode as `ADDR10`.
 `ADDR7` specifies the address mode as `ADDR7`.
- Examples** `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR10` sets the address mode for I2C bus trigger search 1 to `ADDR10`.
- `SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR7`, indicating that the address mode for I2C bus trigger search 2 is set to `ADDR7`.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:I2C:ADDRESS:VALUE

This command sets or queries the binary address string used for the I2C search the specified search condition is `Address` or `AddressData`. The search number is specified by x.

- Conditions** Requires option 2-SERIAL or 2-ULTIMATE.
- Group** Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRESS:VALUE <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRESS:VALUE?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRESS:MODE](#)

Arguments <QString> specifies the address value. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for addresses in hexadecimal format; and 0, 1, and X otherwise.

Examples SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "01XXXXX" sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "XX", indicating that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition

This command sets or queries the search condition for an I2C bus. The search number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition
{ADDRESS|ADDRANDDATA|DATA|ACKMISS|REPEATstart|START|STOP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition?

Arguments ADDRESS specifies the trigger condition as Address.
ADDRANDDATA specifies the trigger condition as Address and Data.
DATA specifies the trigger condition as Data.
ACKMISS specifies the trigger condition as Missing of Acknowledgement.
REPEATstart specifies the trigger condition as Repeat of Start.
START specifies the trigger condition as Start.
STOP specifies the trigger condition as Stop.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRESS` sets the trigger condition for I2C bus trigger search 1 to Address.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRANDDATA`, indicating that the trigger condition for I2C bus trigger search 1 is set to Address and Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:DIRection

This command sets or queries the direction of the data for the I2C bus search to determine where to place a mark. The search number is specified by x. Read or write is indicated by the R/W bit in the I2C protocol.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Search and Mark

Syntax `SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:DIRection`
 `{NOCARE|READ|WRITE}`
 `SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:DIRection?`

Arguments NOCARE specifies the direction of data as Don't Care.

 READ specifies the direction of data as Read.

 WRITE specifies the direction of data as Write.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:DIRECTION READ` sets the data direction for I2C bus trigger search 1 to READ.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION DONTCARE`, indicating that the data direction of the I2C bus trigger search 2 is DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:SIZE

This command sets or queries the length of the data string in bytes used for an I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:SIZE <NR1> SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:SIZE?
Arguments	<NR1> specifies the data size in bytes.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 1 sets the length of the data string for I2C bus trigger search 1 to 1 byte. SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 3, indicating that the length of the data string for I2C bus trigger search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:VALue

This command sets or queries the binary data string used for I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATA:VALue?
Arguments	<QString> specifies the data value. The valid characters are 0, 1, or X for binary format; and A-F, 0-9, and X for hexadecimal format.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:VALUE "1001" sets the data value for I2C bus trigger search 1 to "1001". SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE "XX", indicating that the data value for I2C bus trigger search is "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition

This command sets or queries the condition for a LIN bus search. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition {DATA IDANDDATA ERRor IDentifier SLEEP SYNCfield WAKEup} SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition?
Arguments	Arguments are the available trigger conditions.
Examples	SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition DATA sets the trigger condition to data. SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:CONDITION SYNC, indicating the trigger condition is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:HIVALue

This command sets or queries the high data value string used in a LIN bus search. The search number is specified by x. The search condition must be DATA or IDANDDATA and the data qualifier must be INRANGE or OUTRANGE.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:HIVALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:HIVALue?
Arguments	<QString> is a quoted string of 1s, 0s, or Xs representing the binary data string to be used in a LIN search if the search condition is IDentifier or IDANDDATA (identifier and data).
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE "XX XXXXXXXXXXXX" indicating the high value is "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:QUALifier

This command sets or queries the data qualifier used in a LIN bus search. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:QUALifier {EQUAL LESSEQUAL MOREEQUAL UNEQUAL LESSTHAN MORETHAN INRANGE OUTRANGE} SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:QUALifier?
Arguments	Arguments are the available data qualifiers.
Examples	SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATA:QUALifier LESS sets the data qualifier to less than. SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATA:QUALifier? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:QUALIFIER EQUAL, indicating the data qualifier is equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:SIZE

This command sets or queries the length of the stat string in bytes used for a LIN bus search. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:SIZE <NR1> SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:SIZE?
Arguments	<NR1> specifies the data size.
Examples	SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATA:SIZE 1.0 sets the data size to 1.

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATA:SIZE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:SIZE 1, indicating the data
size is 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:VALue

This command sets or queries the data string used for a LIN bus search. The search number is specified by x. The search condition must be DATA or IDANDDATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATA:VALue?
Arguments	<QString> specifies the data value.
Examples	SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATA:VALue 1010 sets the data value to XXXXXX1010. SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATA:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:VALUE "XXXXXXXXXX", indicating that the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE

This command sets or queries the error type for a LIN bus search. The search number is specified by x. The search condition must be set to ERROR.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE {CHECKsum PARity SYNC} SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE?

Arguments	Checksum specifies the error type is checksum. PARity specifies the error type is parity. SYNC specifies the error type is sync.
Examples	SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE Parity sets the error type to parity. SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE SYNC, indicating that the error type is sync.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:LIN:Identifier:VALue

This command sets or queries the string used for a LIN bus identifier value. The search number is specified by x. The search condition must be IDENTIFIER or IDANDDATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGGER:A:BUS:LIN:Identifier:VALue <QString> SEARCH:SEARCH<x>:TRIGGER:A:BUS:LIN:Identifier:VALue?
Arguments	<QString> specifies the identifier value.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:Identifier:VALue 1010 sets the identifier value to XX1010. SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:Identifier:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:IDENTIFIER:VALUE "XXXXXX", indicates the identifier value is don't care.

SEARCH:SEARCH<x>:TRIGGER:A:BUS:PARAllel:DATA:VALue

This command sets or queries the binary data string used for a parallel bus search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
--------------	-----------------

Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:PARAllel:DATA:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:PARAllel:DATA:VALue?
Arguments	<QString> specifies the data value in a valid format. Valid characters are 0-9,
Examples	SEARCH:SEARCH2:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXX" sets the string data value to "XXXXXXXX" in binary format. SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXXXXXXXXXXXXXX10010", indicating that the data value string is set to "XXXXXXXXXXXXXXXXXXXX10010" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition

This command sets or queries the condition for an RS232C bus search to determine where to place a mark. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition {DATA EOP PARItYerror START} SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition?
Arguments	DATA specifies the search condition as Data. EOP specifies the search condition as End of Packet. PARItYerror specifies the search condition as Parity Error. START specifies the search condition as Start.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:CONDITION DATA sets the search condition for RS232C bus trigger search 1 to Data. SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION EOP, indicating that the search condition for RS232C bus trigger search 2 is set to End of Packet.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:SIZE

This command sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:SIZE <NR3> SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:SIZE?
Arguments	<NR3> is the number of bits per word in the data string, from 1 to 8.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:SIZE 1 sets the number of bits per word in RS232C bus trigger search 1 data string to 1. SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE 8, indicating that the bits per word for RS232C bus trigger search 2 is set to 8.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:VALue

This command sets or queries the data string used for the specified RS232C bus trigger search to determine where to place a mark. The search condition must be Data. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATA:VALue?
Arguments	<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "01" sets the value of the data string for RS232C bus trigger search 1 to "01" when the format is hexadecimal.

SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXXXX1", indicating that the data string value for RS232C bus trigger search 1 is set to "XXXXXXXX1" when the format is binary.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

This command sets or queries the search condition for a SENT bus.

Conditions Requires option SRAUTOSEN.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
{START|FAST|SLOW|PAUSE|ERROR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition?

Arguments Search<x> is the number of the search.

START specifies searching for start of packet.

FAST specifies searching for fast channel data.

SLOW specifies searching for slow channel data.

PAUSE specifies searching for pause pulses.

ERROR specifies searching on errors.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION SLOW specifies searching for slow channel packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION PAUSE, indicating the search is set to find pause pulses.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType

This command sets or queries the error type to be used when searching on SENT data.

Conditions	Requires option SRAUTOSEN. SENT bus trigger condition is set to ERRor.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType {FRAMELENGTH CRC} SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
Arguments	Search<x> is the number of the search. FRAMELENGTH specifies searching for SENT frame length errors. CRC specifies searching for CRC errors.
Examples	SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE FRAMELENGTH sets the search to find SENT frame length errors. SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE? might return SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE CRC to indicate the search is set to find CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRc

This command sets or queries the CRC error type to be used when searching on SENT data.

Conditions	Requires option SRAUTOSEN. SENT bus trigger condition is set to ERRor and ERRType is set to CRc.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRc {FAST SLOW} SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRc?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType

Arguments `Search<x>` is the number of the search.
 `FAST` specifies searching for CRC errors only in the fast channel.
 `SLOW` specifies searching for CRC errors only in the slow channel.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC SLOW` sets the search to find SENT slow channel CRC errors.
 `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC FAST`, indicating that the SENT fast channel is being searched for CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.

Conditions Requires option `SRAUTOSEN`.
 The SENT bus search condition must be set to `FAST`.

Group Trigger

Syntax `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue`
 `<Qstring>`
 `SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue?`

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments `Search<x>` is the Search identifier number.
 `<Qstring>` sets the Fast Channel 1 binary data high value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue`
 `"XXXXXXXXXXXX"` sets the Fast Channel 1 high value on which to search to `XXXXXXXXXXXX`, or "don't care."
 `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALUE`
 `"0101XXXX1111"`, indicating the Fast Channel 1 high value on which to search is set to the binary value `0101XXXX1111`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:
QUALifier {Equal|UNEQUAL|LESSThan|MOREthan|
LESSEQUAL|MOREEQUAL|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

[SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue](#)

Arguments Search<x> is the number of the search.

EQUAL specifies the qualifier as Equal.

LESSEQUAL specifies the qualifier as Less Than or Equal to.

LESSThan specifies the qualifier as Less Than.

MOREEQUAL specifies the qualifier as More Than or Equal to.

MOREThan specifies the qualifier as More Than.

UNEQUAL specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER
UNEQUAL sets the fast channel 1 data qualifier to not equal for search 1.

SEARCH:SEARCH3:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER?
might return

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER
MOREEQUAL to indicate that the fast channel 1 data qualifier is set
to greater than or equal for search 3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments Search<x> is the number of the search.
<Qstring> is the Fast Channel 1 binary value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE
"XXXXXXXXXXXX" sets the Fast Channel 1 value to XXXXXXXXXXXXXXX, or
"don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE? might
return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE
"0000XXXX1111" to indicate the binary value 0000XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

Group Trigger

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue
<Qstring>

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments Search<x> is the Search identifier number.
 <Qstring> sets the Fast Channel 2 high binary data value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue
 "100000000000" sets the Fast Channel 2 high value on which to search to
 100000000000.
 SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue? might
 return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALUE
 "0101XXXX1111", indicating the Fast Channel 2 high value on which to search is
 set to the binary value 0101XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.

Conditions Requires option SRAUTOSEN
 The SENT bus trigger condition must be set to FAST.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:
 QUALifier {EQua1|UNEQua1|LESSthan|MORethan|
 LESSEQua1|MOREEQua1|INrange|OUTrange}
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)
[SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue](#)

Arguments Search<x> is the number of the search.
 EQua1 specifies the qualifier as Equal.
 LESSEQua1 specifies the qualifier as Less Than or Equal to.
 LESSThan specifies the qualifier as Less Than.

MOREEqual specifies the qualifier as More Than or Equal to.

MOREThan specifies the qualifier as More Than.

UNEQUAL specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER
UNEQUAL sets the fast channel 2 data qualifier to not equal for search 5.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER?
might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER
MOREEQUAL to indicate that the fast channel 2 data qualifier is set
to greater than or equal for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
The trigger condition must be set to FAST.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments search<x> is the number of the search.
<Qstring> is the Fast Channel 2 binary value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE
"111111111111" sets the Fast Channel 2 value to 111111111111.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE "000000000000" to indicate the binary value 000000000000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:HIVALue

This command sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group Trigger

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:HIVALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:HIVALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)
[BUS:B<x>:SENT:NUMCHANnel](#)
[BUS:B<x>:SENT:NIBBLECount](#)

Arguments Search<x> is the Search identifier number.
<Qstring> sets the Fast Channel 1 counter binary value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE
"XXXXXXXX" sets the Fast Channel secure counter high value on which to search to "don't care."

SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE? might return SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE "11110000" to indicate a binary value of 11110000 on which to search.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.

Conditions	Requires option SRAUTOSEN The SENT bus trigger condition must be set to FAST.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:QUALifier {Equal UNEQual LESSthan MOREthan LESSEQual MOREEQual INrange OUTrange} SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:QUALifier?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:VALue
Arguments	Search<x> is the number of the search. EQual specifies the qualifier as Equal. LESSEQual specifies the qualifier as Less Than or Equal to. LESSThan specifies the qualifier as Less Than. MOREEQual specifies the qualifier as More Than or Equal to. MOREThan specifies the qualifier as More Than. UNEQual specifies the qualifier as Unequal. INrange sets the qualifier to inside a range. OUTrange sets the qualifier to outside a range.
Examples	SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:COUNTER:QUALifier LESSThan sets the fast channel 2 data qualifier on search 5 to less than. SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:QUALifier? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:QUALIFIER EQUAL to indicate that the fast channel 2 data qualifier on search 1 is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:VALue

This command sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.

Conditions	Requires option SRAUTOSEN. The search condition must be set to FAST. The number of channels must be set to 1. The nibble count must be set to 6.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:VALue <Qstring> SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTER:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)
[BUS:B<x>:SENT:NIBBLECount](#)
[BUS:B<x>:SENT:NUMCHANnel](#)

Arguments Search<x> is the number of the search.
<Qstring> is the Fast Channel 1 counter value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE
"XXXXXXXX" sets the Fast Channel 1 secure counter value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE? might
return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE
"00001111" to search for a binary counter value of 00001111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
The search condition must be set to FAST.
The number of channels must be set to 1.

The nibble count must be set to 6.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)
[BUS:B<x>:SENT:NIBBLECount](#)
[BUS:B<x>:SENT:NUMCHANnel](#)

Arguments Search<x> is the number of the search.
<Qstring> is the fast message inverted nibble binary value.

Examples TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX" sets the Fast Channel 1 secure inverted nibble value to "don't care."
TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "1111" to search for a binary inverted nibble value of 1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
The trigger condition must be set to FAST.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments	Search<x> is the number of the search. <Qstring> is the binary status binary value.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE "XXXX" sets the binary status value to XXXX. SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE "1010" to indicate searching on a binary status value of 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when searching on SENT pause pulses.

Conditions	Requires option SRAUTOSEN The SENT bus trigger condition must be set to PAUSE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE: QUALifier {Equal UNEQual LESSthan MORethan LESSEQual MOREEQual INrange OUTrange} SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue
Arguments	Search<x> is the number of the search. EQual specifies the qualifier as Equal. INrange sets the qualifier to be within a range. LESSEQual sets the qualifier as Less Than or Equal to. LESSThan sets the qualifier as Less Than. MOREEQual sets the qualifier as More Than or Equal to. MORETHan sets the qualifier as More Than.

OUTrange sets the qualifier to be outside a range.

UNEQUAL specifies the qualifier as Unequal.

- Examples** SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER LESSTHAN sets the pause pulse qualifier to less than.
- SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER INRANGE to indicate that the pause pulse qualifier is set to search inside the range of ticks low and ticks high.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

This command sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.

- Conditions** Requires option SRAUTOSEN.
The trigger condition must be set to PAUSE.
- Group** Search and Mark
- Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue ?
- Related Commands** [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)
- Arguments** Search<x> is the number of the search.
<NR1> is the maximum number of pause clock ticks to be used when searching.
- Examples** SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKs:HIVALUE 15 sets the maximum pause tick count for which to search to 15.
- SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKs:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKs:HIVALUE 20 to indicate the maximum number of pause ticks to search for is 20.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

This command sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions	Requires option SRAUTOSEN. The trigger condition must be set to PAUSE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKS:VALue <NR1> SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKS:VALue?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
Arguments	Search<x> is the number of the search. <NR1> is the minimum number of pause clock ticks to be used when searching.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 12 sets the minimum pause tick count on which to search to 12. SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 15 to indicate the minimum number of pause ticks to search for is 15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when searching on a SENT bus signal.

Conditions	Requires option SRAUTOSEN. The SENT bus search condition must be set to SLOW.
Group	Trigger
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments	Search<x> is the Search identifier number. <Qstring> sets the binary Slow channel data value.
Examples	SEARCH:SEARCH2:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "XXXXXXXX" sets the Slow data high value on which to search to "don't care" for Search 2. SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "11110001", to indicate searching on slow data values that match 11110001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier

This command sets or queries the qualifier to be used when searching on SENT slow packet bus data.

Conditions	Requires option SRAUTOSEN. The SENT bus trigger condition must be set to SLOW.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA: QUALifier {Equal UNEQual LESSthan MOREthan LESSEQual MOREEQual INrange OUTrange} SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue
Arguments	Search<x> is the number of the search. EQual specifies the qualifier as Equal. LESSEQual sets the qualifier as Less Than or Equal to. LESSThan sets the qualifier as Less Than. MOREEQual sets the qualifier as More Than or Equal to. MOREThan sets the qualifier as More Than. UNEQual specifies the qualifier as Unequal. INrange sets the search qualifier to inside a range.

OUTrange sets the search qualifier to outside a range.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER
LESSEQUAL sets the slow channel data qualifier to less than or equal.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER? might
return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER
EQUAL to indicate that the slow channel data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue

This command sets or queries the binary slow channel data value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
The trigger condition must be set to SLOW.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition](#)

Arguments Search<x> is the number of the search.
<Qstring> is the slow channel data binary value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE "XXXXXXXX"
sets the slow data search value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE? might return
"XXXX1111", indicating the search is marking slow data values that match
XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue

This command sets or queries the binary slow identifier value to be used when searching on a SENT bus signal.

Conditions	Requires option SRAUTOSEN. The trigger condition must be set to SLOW.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue <Qstring> SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue?
Related Commands	SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
Arguments	search<x> is the number of the search. <Qstring> is the slow channel identifier binary value.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "XXXX" sets the search identifier value to "don't care." SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "0001" indicating to search for the slow channel identifier value of 0001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce

This command sets or queries the bus source for the bus search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce {B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16} SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce?
Arguments	B0–B16 specifies the bus source as a bus number from B01 to B16.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B1 sets the bus source for bus trigger search 1 to B1.

SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B15, indicating that the bus
source for bus trigger search 1 is B15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition

This command sets or queries the search condition for an SPI bus search to determine where to place a mark. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition {DATA SS STARTofframe} SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition?
Arguments	DATA specifies the trigger condition as Data. SS specifies the trigger condition as Slave Selection. STARTofframe specifies the trigger condition as start of frame.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION SS sets the trigger condition for SPI bus trigger search 1 to SS. SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION DATA, indicating that the trigger condition for SPI bus trigger search 1 is set to Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:SIZE

This command sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark. The search condition must be DATA. The search number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark

Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:SIZE <NR1> SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:SIZE?
Arguments	<NR1> specifies the number of contiguous data bytes.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:SIZE 1 sets the length of the data string for SPI bus trigger search 1 to 1 byte. SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE 4, indicating that the length of the data string for SPI bus trigger search 2 is 4 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:VALue

The command sets or queries the binary data string used for an SPI bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:VALue <QString> SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:VALue?
Arguments	<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; and A-F, 0-9, and X for hexadecimal format.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "00001111" sets the data value for SPI bus trigger search 1 to "00001111" in binary format. SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "11", indicating that the data value for SPI bus trigger search 1 is set to "11" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCEType

This command sets or queries trigger Source for SPI bus. The search number is specified by x.

Conditions	Requires option SR-SPI.
Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCEType {MISo MOSi} SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCEType?
Arguments	MISo specifies the trigger source as MISo. The default search source type is MISo. MOSi specifies the trigger source as MOSi.
Examples	SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCEType MOSi sets the trigger source for SPI bus trigger search 1 to MOSi. SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCEType? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCEType MOSi, indicating that the trigger source for SPI bus trigger search 1 is MOSi.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe

This command sets or queries the slope for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe {RISe FALL EITHer} SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe?
Arguments	RISe specifies a rising edge. FALL specifies a falling edge. EITHer specifies either rising or falling edge.
Examples	SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISe sets the slope for search 1 to rise. SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISe, indicating that the slope for the trigger for search 1 is rise.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce

This command sets or queries the source waveform for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce {CH<x>| DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce?

Arguments CH<x> specifies one input channel as the edge source, where the channel number is specified by x.

DCH<x>_D<x> specifies a digital waveform as the source waveform for the specified search. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> specifies the math waveform as the search source, where the math number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.

Examples SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2 sets the source waveform for the A trigger to Channel 2.

SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE? might return
SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2, indicating that the Channel 2 edge trigger is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold

This command sets or queries the source threshold level for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold?

Arguments	<NR3> is the source threshold level for an edge trigger search.
Examples	<pre>SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 50.0e-3</pre> sets the threshold to 50 mV. <pre>SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD?</pre> might return <pre>SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 0.0E+0</pre> indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:CLOCK:THRESHOLD

This command sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	<pre>SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:CLOCK:THRESHOLD <NR3></pre> <pre>SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:CLOCK:THRESHOLD?</pre>
Arguments	<NR3> is the logic clock threshold.
Examples	<pre>SEARCH:SEARCH1:TRIGGER:A:LOGIC:CLOCK:THRESHOLD 50.0e-3</pre> sets the threshold to 50 mV. <pre>SEARCH:SEARCH1:TRIGGER:A:LOGIC:CLOCK:THRESHOLD?</pre> might return <pre>SEARCH:SEARCH1:TRIGGER:A:LOGIC:CLOCK:THRESHOLD 0.0E+0</pre> indicating the threshold is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:DELTAtime

This command specifies the Logic search delta time value. The time value is used as part of the Logic search condition to determine if the duration of a logic pattern meets the specified time constraints. The search number is specified by x.

Group	Search and Mark
Syntax	<pre>SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:DELTAtime <NR3></pre> <pre>SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:DELTAtime?</pre>
Arguments	<NR3> is delta time value.

- Examples** `SEARCH:SEARCH1:TRIGger:A:LOGic:DELTatime 1.5e-9` sets the delta time to 1.5 ns.
- `SEARCH:SEARCH1:TRIGger:A:LOGic:DELTatime?` might return `SEARCH:SEARCH1:TRIGGER:A:LOGIC:DELTATIME 1.0000E-9` indicating the delta time is 1.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNcTion

This command sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark. The search number is specified by x.

- Group** Search and Mark
- Syntax** `SEARCH:SEARCH<x>:TRIGger:A:LOGic:FUNcTion {AND|NAND|NOR|OR}`
`SEARCH:SEARCH<x>:TRIGger:A:LOGIC:FUNcTion?`
- Arguments** AND places a mark if all conditions are true.
 NAND places a mark if any of the conditions are false.
 NOR places a mark if all conditions are false.
 OR places a mark if any of the conditions are true.
- Examples** `SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION AND` sets the trigger a logic function for search 1 to AND.
- `SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION?` might return `SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION NOR`, indicating that the logic function for search 1 is set to NOR.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce

This command specifies or queries the channel to use as the clock source for logic trigger. The search number is specified by x.

- Group** Search and Mark
- Syntax** `SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPUT:CLOCK:SOUrce {CH<x>|`
`DCH<x>_D<x>| REF<x>_D<x>}`
`SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPUT:CLOCK:SOUrce?`

Arguments	<p>CH<x> specifies one input channel as the edge source, where the channel number is specified by x.</p> <p>DCH<x>_D<x> specifies a digital waveform as the source waveform for the specified search. The supported digital channel value is 1. The supported digital bit values are 0 to 15.</p> <p>REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.</p>
Examples	<p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE CH2 sets the clock source to channel 2.</p> <p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE UNDEFINED indicating the clock source is not defined.</p>

SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:CH<x>

This command sets or queries the voltage level to use for logic trigger search. The search number is specified by x.

Group	Search and Mark
Syntax	<pre>SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:CH<x>?</pre>
Arguments	<NR3> is the voltage level to use for logic trigger search.
Examples	<p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:CH2 50.0e-3 sets the level to 50 mV.</p> <p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:CH2? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:CH2 0.0E+0 indicating the level is set to 0.0 V.</p>

SEARCH:SEARCH<x>:TRIGGER:A:LOGIC:LEVEL:MATH<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by x.

Group	Search and Mark
--------------	-----------------

Syntax	SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVe1:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVe1:MATH<x>?
Arguments	<NR3> is the voltage level to use for logic trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:LOGic:LEVe1:MATH1 50.0e-3 sets the level to 50.0 mV. SEARCH:SEARCH1:TRIGger:A:LOGic:LEVe1:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:MATH1 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVe1:REF<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVe1:REF<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:LOGic:LEVe1:REF<x>?
Arguments	<NR3> is the voltage level to use for logic trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:LOGic:LEVe1:REF1 50.0e-3 sets the level to 50.0 mV. SEARCH:SEARCH1:TRIGger:A:LOGic:LEVe1:REF1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:REF1 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:CH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group	Search and Mark
--------------	-----------------

Syntax SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:CH<x> {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:CH<x>?

Arguments H specifies triggering when the pattern is high.
L specifies triggering when the pattern is low.
X specifies triggering when the pattern is high or low.

Examples SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:CH2 H sets the channel 2 pattern to a high.
SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:CH2? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:CH2 X indicating channel 2 is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:DCH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:DCH<x>_D<x>
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGic:LOGICPattern:DCH<x>_D<x>?

Arguments H specifies triggering when the pattern is high.
L specifies triggering when the pattern is low.
X specifies triggering when the pattern is high or low.

Examples SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:DCH1_D1 H sets the channel pattern to a high.
SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:DCH1_D1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:DCH1_D1 X indicating channel 2 is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x> {H L X} SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>?
Arguments	H specifies triggering when the pattern is high. L specifies triggering when the pattern is low. X specifies triggering when the pattern is high or low.
Examples	SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:MATH1 H sets the pattern to a high. SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:MATH1 X indicating the pattern is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x> {H L X} SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>?
Arguments	H specifies triggering when the pattern is high. L specifies triggering when the pattern is low. X specifies triggering when the pattern is high or low.

Examples SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:REF1 H sets the pattern to a high.
SEARCH:SEARCH1:TRIGger:A:LOGic:LOGICPattern:REF1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:REF1 X indicating the pattern is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGic:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:LOGic:POLarity {POSitive|NEGative|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:LOGic:POLarity?

Arguments POSitive specifies using the positive clock edge.
NEGative specifies using negative clock edge.
EITHer specifies using either the positive or negative clock edge.

Examples SEARCH:SEARCH1:TRIGger:A:LOGic:POLarity NEGATIVE sets the polarity to negative.
SEARCH:SEARCH1:TRIGger:A:LOGic:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:POLARITY POSITIVE indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:LOGic:USEClOCKedge

This command specifies whether or not Logic search uses a clock source. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:LOGic:USEClOCKedge {OFF|ON|0|1}
SEARCH:SEARCH<x>:TRIGger:A:LOGic:USEClOCKedge?

Arguments	<p>OFF specifies not to use the clock source.</p> <p>ON specifies to use the clock source.</p> <p>0 specifies not to use the clock source.</p> <p>1 specifies to use the clock source.</p>
Examples	<p>SEARCH:SEARCH1:TRIGger:A:LOGic:USEClockedge ON specifies to use the clock source.</p> <p>SEARCH:SEARCH1:TRIGger:A:LOGic:USEClockedge? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:USECLOCKEDGE 0 indicating not to use the clock source.</p>

SEARCH:SEARCH<x>:TRIGger:A:LOGic:WHEn

This command sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.

Group	Search and Mark
Syntax	<pre>SEARCH:SEARCH<x>:TRIGger:A:LOGic:WHEn {TRUE FALSE MOREthan LESSThan EQua1 UNEQua1} SEARCH:SEARCH<x>:TRIGger:A:LOGic:WHEn?</pre>
Arguments	<p>TRUe searches on an input value that is true.</p> <p>FALSe searches on an input value that is false.</p> <p>MOREthan searches on an input value that is greater than a set value.</p> <p>LESSThan searches on an input value that is less than a set value.</p> <p>EQua1 searches on an input value that is equal to a set value.</p> <p>UNEQua1 searches on an input value that is not equal to a set value.</p>
Examples	<p>SEARCH:SEARCH1:TRIGger:A:LOGic:WHEn FALSE specifies a search on an input value that is false.</p> <p>SEARCH:SEARCH1:TRIGger:A:LOGic:WHEn? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:WHEN TRUE indicating a search on an input value that is true.</p>

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:HIGHLimit <NR3> SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:HIGHLimit?
Arguments	<NR3> is the upper limit to use, in seconds, when searching for a pulse.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:HIGHLimit 2.5e-9 sets the high limit to 2.5 ns. SEARCH:SEARCH1:TRIGger:A:PULSEwidth:HIGHLimit? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT 2.0000E-9 indicating the pulse width high limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOGICQUALification {ON OFF} SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOGICQUALification?
Arguments	ON specifies to use logic qualification. OFF specifies not to use logic qualification.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:LOGICQUALification ON turns on logic qualification. SEARCH:SEARCH1:TRIGger:A:PULSEwidth:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOWLimit <NR3> SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:LOWLimit?
Arguments	<NR3> is the lower limit to use, in seconds, when searching for a pulse.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:LOWLimit 0.5e-9 sets the low limit to 0.5 ns. SEARCH:SEARCH1:TRIGger:A:PULSEwidth:LOWLimit? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT 1.0000E-9 indicating the low limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity

This command specifies the polarity for a pulse width search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity {POSitive NEGative} SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity?
Arguments	POSitive specifies positive polarity for a pulse width search. NEGative specifies negative polarity for a pulse width search.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:POLarity NEGATIVE sets the polarity to negative. SEARCH:SEARCH1:TRIGger:A:PULSEwidth:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY POSITIVE indicating the polarity is positive.

SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce

This command sets and queries the source for the pulse width search input. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce {CH<x> DCH<x>_D<x> REF<x> REF<x>_D<x>} SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce?
Arguments	CH<x> specifies an analog channel to use as the source. DCH<x>_D<x> specifies a digital channel to use as the source. The supported digital channel value is 1. The supported digital bit values are 0 to 15. MATH<x> specifies a math waveform to use as the source. REF<x> specifies a reference waveform to use as the source. REF<x>_D<x> specifies a digital reference waveform as the source.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:SOURce CH1 sets the source to channel 1. SEARCH:SEARCH1:TRIGger:A:PULSEwidth:SOURce? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:SOURCE CH2 indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold

Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold <NR3> SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:THReshold?
Arguments	<NR3> is the source threshold level for a pulse width trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:PULSEwidth:THReshold 1.0e-9 sets to 1.0 V.

SEARCH:SEARCH1:TRIGger:A:PULSEwidth:THRESHold? might return
 SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:THRESHOLD 0.0E+0 indicating
 the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn

This command specifies to search for a pulse with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using SEARCH:A:PULSEwidth:WIDTH), OR whose SEARCH:A:PULSEwidth:LOWLimit and SEARCH:A:PULSEwidth:HIGHLimit). The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn
 {LESSthan|MOREthan| EQual|UNEQual|WITHin|OUTside}
 SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn?

Arguments LESSthan causes a search when a pulse is detected with a width less than the time set by the SEARCH:A:PULSEWidth:WIDTH command.

MOREthan causes a search when a pulse is detected with a width greater than the time set by the SEARCH:A:PULSEWidth:WIDTH command.

EQual causes a search when a pulse is detected with a width equal to the time period specified in SEARCH:A:PULSEWidth:WIDTH within a $\pm 5\%$ tolerance.

UNEQual causes a search when a pulse is detected with a width greater than or less than (but not equal) the time period specified in SEARCH:A:PULSEWidth:WIDTH within a $\pm 5\%$ tolerance.

WITHin causes a search when a pulse is detected that is within a range set by two values.

OUTside causes a search when a pulse is detected that is outside of a range set by two values.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEwidth:WHEn Outside causes a search when a pulse is detected that is outside the set range.

SEARCH:SEARCH1:TRIGger:A:PULSEwidth:WHEn? might return
 SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN WITHIN indicating that a search will occur when a pulse is detected that is within a set range.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification

This command specifies whether or not to use logic qualification for a runt search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification {ON OFF} SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification?
Arguments	ON specifies to use logic qualification for a runt search. OFF specifies not to use logic qualification for a runt search.
Examples	SEARCH:SEARCH1:TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification. SEARCH:SEARCH1:TRIGger:A:RUNT:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity

This command specifies the polarity for the runt search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity {POSitive NEGative EITher} SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?
Arguments	POSitive specifies using positive polarity for the runt search. NEGative specifies using negative polarity for the runt search. EITher specifies using either positive or negative polarity for the runt search.
Examples	SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGATIVE specifies using negative polarity for the runt search.

SEARCH:SEARCH1:TRIGger:A:RUNT:POLArity? might return
 SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY POSITIVE indicating the
 polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce

This command sets and queries the source for the Runt search input. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce {CH<x> REF<x>}
Arguments	Arguments are the available sources.
Examples	SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce CH1 sets the source to channel 1. SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:SOURCE CH2 indicating the source is set to channel 2.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh

This command sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh <NR3> SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh?
Arguments	<NR3> is the source threshold HIGH level for a runt trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGh 50.0E-3 sets the high threshold to 50 mV. SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGh? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW

Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW <NR3> SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW?
Arguments	<NR3> is the source threshold LOW level for a runt trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW 50.0e-3 sets the threshold to 50 mV. SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

This command sets or queries the condition setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn {OCCURS LESSthan MOREthan EQua1}NOTEQua1} SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?
Arguments	LESSthan argument sets the instrument to search if the a runt pulse is detected with width less than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTh command. MOREthan argument sets the instrument to search if the a runt pulse is detected with width more than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTh command. EQua1 argument sets the instrument to search when the pattern is true for a time period equal to the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTh within a $\pm 5\%$ tolerance.

NOTEEqual argument sets the instrument to search when the pattern is true for a time period greater than or less than (but not equal) the time period specified in SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH within a $\pm 5\%$ tolerance.

OCCURS argument specifies a search event if a runt of any detectable width occurs.

Examples SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN MORETHAN sets the instrument to trigger when a runt pulse is detected with width wider than the time set by the SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH command.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN OCCURS, indicating that a mark is placed if a runt trigger event occurs.

SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH

This command sets or queries the width setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH <NR3>
SEARCH:SEARCH<x>:TRIGGER:A:RUNT:WIDTH?

Arguments <NR3> specifies the minimum width in seconds.

Examples SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 400E-12 sets the runt trigger minimum width for search 1 to 0.4 nanoseconds.
SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 500.0000E-12, indicating that the runt trigger minimum width for search 1 is set to 0.5 nanoseconds.

SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE

This command sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE {FALL|RISE}
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:CLOCK:EDGE?

Arguments FALL specifies the polarity as the clock falling edge.
RISe specifies the polarity as the clock rising edge.

Examples SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE FALL sets the setup/hold trigger clock slope polarity for search 1 to falling edge.
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE RISE, indicating that the setup/hold trigger clock slope polarity for search 1 is set to rising edge.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce

This command sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce {CH<x>|DCH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce?

Arguments CH<x> specifies an input channel as the edge source, where <x> = 1, 2, 3, 4, 5, 6, 7, or 8, depending on the number of channels in your instrument.
DCH<x>_D<x> specifies a digital waveform as the setup and hold clock source waveform for the specified search. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
MATH<x> specifies the math waveform as the search source, where <x> = ≥1.
REF<x> specifies the reference waveform as the search source, where <x> = ≥1.
REF<x>_D<x> specifies a digital reference waveform as the setup and hold clock source waveform for the specified search.

Examples SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE MATH1 sets the setup/hold trigger clock source setting for search 1 to MATH1.
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1, indicating that the setup/hold trigger clock source setting for search 1 is set to CH1.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold?
Arguments	<NR3> the clock threshold setting for a setup/hold trigger search.
Examples	SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3 sets the setup/hold trigger clock threshold setting for search 1 to -1.3 volts. SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3000, indicating that the setup/hold trigger clock threshold setting for search 1 is set to -1.3 volts.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime

This command sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime?
Arguments	<NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.
Examples	SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 400E-12 sets the setup/hold trigger hold time setting for search 1 to 400 ps. SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 500.0000E-12, indicating that the setup/hold trigger hold time setting for search 1 is set to 0.5 ns.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>?
Arguments	<NR3> the voltage level to use for setup & hold trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1 50.0e-3 sets the level to 50.0 mV. SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:CH1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>?
Arguments	<NR3> is the voltage level to use for setup & hold trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1 50.0e-3 sets the level to 50.0 mV. SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:MATH1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>?
Arguments	<NR3> is the voltage level to use for setup & hold trigger search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1 50.0e-3 sets the level to 50.0 mV. SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:REF1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x> {INCLUDE DONTInclude} SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>?
Arguments	INCLUDE specifies including the specified channel SETHOLD inputs in the specified search. DONTInclude specifies not including the specified channel SETHOLD inputs in the specified search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:CH1 INCLUDE specifies including the specified channel SETHOLD inputs in the specified search.

SEARCH:SEARCH1:TRIGGER:A:SETHold:LOGICPattern:CH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:CH1 DONTINCLUDE indicating the specified channel SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGGER:A:SETHold :LOGICPattern:DCH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGGER:A:SETHold :LOGICPattern:DCH<x>_D<x> {INCLUDE DONTINCLUDE} SEARCH:SEARCH<x>:TRIGGER:A:SETHold :LOGICPattern: DCH<x>_D<x>?
Arguments	INCLUDE specifies including the specified channel SETHOLD inputs in the specified search. DONTINCLUDE specifies not including the specified channel SETHOLD inputs in the specified search.
Examples	SEARCH:SEARCH1:TRIGGER:A:SETHold:LOGICPattern:DCH1_D0 INCLUDE specifies including the specified digital channel SETHOLD inputs in the specified search. SEARCH:SEARCH1:TRIGGER:A:SETHold:LOGICPattern:DCH1_D0? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:DCH1_D0 DONTINCLUDE indicating the specified digital channel SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGGER:A:SETHold:LOGICPattern:MATH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x> {INCLude DONTInCLude} SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>?
Arguments	INCLude specifies including the specified math SETHOLD inputs in the specified search. DONTInCLude specifies not including the specified math SETHOLD inputs in the specified search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1 INCLude specifies including the specified math SETHOLD inputs in the specified search. SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:MATH1 DONTINCLUDE indicating the specified math SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>

This command sets and returns the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x> {INCLude DONTInCLude} SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>?
Arguments	INCLude specifies including the specified reference SETHOLD inputs in the specified search. DONTInCLude specifies not including the specified reference SETHOLD inputs in the specified search.
Examples	SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:REF1 INCLude specifies including the specified reference SETHOLD inputs in the specified search. SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:REF1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:REF1

DONTINCLUDE indicating the specified reference SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime

This command sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime?
Arguments	<NR3> specifies the setup time for setup and hold violation triggering.
Examples	SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME 2E-9 sets the setup/hold trigger setup time setting for search 1 to 2 ns. SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME 1.0000E-9, indicating that the setup/hold trigger setup time setting for search 1 is set to 1.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:STATE

This command sets or queries the enabled state of the search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:STATE {<NR1> OFF ON}
Arguments	<NR1> = 1 enables the search. Any other character disables the search. ON enables the search. OFF disables the search.
Examples	SEARCH:SEARCH1:TRIGger:A:STATE OFF disables the search. SEARCH:SEARCH1:TRIGger:A:STATE? might return SEARCH:SEARCH1:TRIGGER:A:STATE 1 indicating search 1 is enabled.

SEARCH:SEARCH<x>:TRIGger:A:STOPAcq

This command sets or queries whether acquisitions are stopped when a search hit is found. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:STOPAcq {<NR1> OFF ON} SEARCH:SEARCH<x>:TRIGger:A:STOPAcq?
Arguments	<x> is the number of the search on which to enable or disable the stop acquisition function. <NR1> = 1 enables stopping when a search hit is found. Any other character disables the feature. ON enables stopping when a search hit is found. OFF disables stopping on a search hit.
Examples	SEARCH:SEARCH1:TRIGger:A:STOPAcq ON enables stopping when a search hit is found. SEARCH:SEARCH3:TRIGger:A:STOPAcq? might return :SEARCH:SEARCH3:TRIGGER:A:STOPACQ 1 indicating acquisitions are stopped when a search hit is found on search number 3.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout search. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification {ON OFF} SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification?
Arguments	ON specifies to use logic qualification. OFF specifies not to use logic qualification.

- Examples** `SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification ON` specifies to use logic qualification.
- `SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:LOGICQUALIFICATION OFF` indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity

The polarity to be used for a Timeout search. The search number is specified by x.

- Group** Search and Mark
- Syntax** `SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity`
 `{STAYSHigh|STAYSLow|EITHer}`
 `SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity?`
- Arguments** `STAYSHigh` specifies the polarity stays HIGH.
 `STAYSLow` specifies the polarity stays LOW.
 `EITHer` specifies the polarity stays HIGH or stays LOW.
- Examples** `SEARCH:SEARCH1:TRIGger:A:TIMEOut:POLarity STAYSLow` specifies the polarity stays LOW.
- `SEARCH:SEARCH1:TRIGger:A:TIMEOut:POLarity?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY STAYSHIGH` indicating the polarity stays HIGH.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce

This command sets and queries the source for timeout search input. The search number is specified by x.

- Group** Search and Mark
- Syntax** `SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce {CH<x>|`
 `DCH<x>_D<x>| MATH<x>| REF<x>| REF<x>_D<x>}`
 `SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce?`

Arguments CH<x> specifies an input channel as the edge source, where <x> = 1, 2, 3, 4, 5, 6, 7, or 8, depending on the number of channels in your instrument.

DCH<x>_D<x> specifies a digital channel for the search source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.

MATH<x> specifies the math waveform as the search source, where <x> = ≥ 1 .

REF<x> specifies the reference waveform as the search source, where <x> = ≥ 1 .

REF<x>_D<x> specifies a digital reference waveform as the source waveform the specified search.

Examples SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOURce

SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOURce? might return
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:SOURCE CH1 indicating CH1 is the source.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold

Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold?

Arguments <NR3> is the source threshold level for a timeout trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold 50.0e-3 sets the threshold to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold? might return
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME

This command sets or queries the time setting for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax	SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME <NR3> SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME?
Arguments	<NR3> is the time in seconds.
Examples	SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the timeout trigger time setting for search 1 to 400 ns. SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME? might return SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 500.0000E-12, indicating that the timeout trigger time setting for search 1 is set to 500 ns.

SEARCH:SEARCH<x>:TRIGger:A:TYPE

This command sets or queries the trigger type setting for a search to determine where to place a mark. The search number is specified by x.

Group	Search and Mark
Syntax	SEARCH:SEARCH<x>:TRIGger:A:TYPE {EDGE RUnt TRAnSition PULSEwidth TIMEOut LOGic SETHold Bus} SEARCH:SEARCH<x>:TRIGger:A:TYPE?
Arguments	EDGE triggers when the source input signal amplitude crosses the specified level in the direction given by the slope.

NOTE. *Some trigger types are optional.*

RUNt triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the SEARCH:SEARCH<x>:TRIGger:A:RUNt:HIGH and SEARCH:SEARCH<x>:TRIGger:A:RUNt:LOW THRESHOLD commands.

TRAnSition triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

PULSEwidth triggers on input signal source pulses that are inside or outside of the given time range specified by SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit and SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit. The polarity is selected using the SEARCH:SEARCH<x>:TRIGger:A:RUNt:POLarity command.

TIMEOut triggers on an input signal source that stays above, stays below, or stays either above or below the trigger level for a given time.

LOGIC specifies that a search occurs when specified conditions are met, and is controlled by the **SEARCH:A:LOGIC** commands.

SETHold triggers on a functional pattern combination of one to three data sources at the time of the clock transition.

Bus specifies that a search occurs when a communications signal is found.

Examples **SEARCH:SEARCH1:TRIGGER:A:TYPE RUNT** sets the trigger type setting for search 1 to RUNT.

SEARCH:SEARCH1:TRIGGER:A:TYPE? might return **SEARCH:SEARCH1:TRIGGER:A:TYPE EDGE**, indicating that the trigger type setting for search 1 is set to EDGE.

SEARCH:SElected

This command sets or queries the selected search, for example **SEARCH1**. The search number is specified by **x**.

Group Search and Mark

Syntax **SEARCH:SElected SEARCH1**

Arguments **SEARCH1** is the specified search.

Examples **SEARCH:SElected SEARCH1** specifies a search 1 search.

SEARCH:SElected? might return **:SEARCH:SELECTED SEARCH1** indicating search 1 is selected.

SElect:DCH<x>:DA11

This command turns on or off all constituent digital channels.

Group Digital

Syntax **SElect:DCH<x>:DA11 {ON|OFF|<NR1>}**
SElect:DCH<x>:DA11?

Arguments DCH<x> specifies the digital channel. The supported value is 1.
 <NR1> = 0 disables the specified digital channel; any other value turns the on
 the digital channel.
 OFF turns off the specified digital channel.
 ON turns on the specified digital channel.

Examples SElect:DCH1:DA11 ON turns ON all the bits of separate digital channel 1.
 SElect:DCH1:DA11? might return :SElect:DCH1:DA11 0 indicating all the
 bits are turned off.

SET? (Query Only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. The SET? query always returns command headers, regardless of the setting of the [HEADer](#) command. This is because the returned commands are intended to be sent back to the instrument as a command string. The [VERBose](#) command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the [*LRN?](#) command.

Group Miscellaneous

Syntax SET?

Related Commands [HEADer](#)
 [*LRN?](#)
 [VERBose](#)

Examples SET? might return the following response: :ACQUIRE:STOPAFTER
 RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET
 1;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE
 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT
 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE
 RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE
 NOTRUNNING;:AUXOUT:SOURCE ATRIGGER;EDGE FALLING;:CMD BATCH
 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE
 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE


```
NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;:DISPLAY:FILTER
SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM
75.0000;AUTOBRIGHT 1;:DISPLAY:PERSISTENCE
OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST
500.0000E-3;PORT FILE;:DIAG:LEVEL SUBSYS;:TRIGGER:A:MODE
AUTO;TYPE EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME
1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;.
```

SOCKETServer:ENABle

This command enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument. The default state is enabled.

Group Miscellaneous

Syntax SOCKETServer:ENABle {0|1|OFF|ON}
SOCKETServer:ENABle?

Arguments 1 enables the socket server. If the state is 0 (disabled) and this command is sent to enable the socket server when the port is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains disabled. In this case, select a different port number and attempt to enable the socket server again.

0 disables the socket server.

ON enables the socket server.

OFF disables the socket server.

Examples SOCKETSERVER:ENABLE On enables the socket server.

SOCKETSERVER:ENABLE? might return :SOCKETSERVER:ENABLE 1 indicating the socket server is enabled.

SOCKETServer:PORT

This command sets the TCPIP port for the socket server connection.

Group Miscellaneous

Syntax SOCKETServer:PORT <NR1>
SOCKETServer:PORT?

Arguments <NR1> is the TCPIP port for the socket server connection.

If the socket server is enabled and the port specified is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains in its current state (i.e. enabled/disabled and port address remain unchanged).

Similarly, if any sessions are active when the port is changed, this same error event may be posted to the event queue and the port will remain unchanged. In this case, exit all current sessions and send the :SOCKETServer:PORT command again.

Examples SOCKETSERVER:PORT 4000 sets the socket server port number to 4000.

SOCKETSERVER:PORT? might return :SOCKETSERVER:PORT 4000 indicating that the port number is 4000.

SOCKETServer:PROTOCOL

This command sets or queries the protocol for the socket server.

Group Miscellaneous

Syntax SOCKETServer:PROTOCOL {TERMiNa|NONE}
SOCKETServer:PROTOCOL?

Arguments TERMiNa specifies terminal protocol for the socket server. When set to TERMiNa, a session startup message is sent to the socket and a command prompt is provided.

NONE disables the terminal features, allowing the server to be used for raw socket transactions, such as with a VISA socket server. The default setting is NONE.

Examples SOCKETSERVER:PROTOCOL NONE sets the protocol to none.

SOCKETSERVER:PROTOCOL? might return :SOCKETSERVER:PROTOCOL TERMINAL indicating the protocol is set to terminal.

***SRE**

The *SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

Group	Status and Error
Syntax	*SRE <NR1> *SRE?
Related Commands	*CLS DESE *ESE *ESR? EVENT? EVMsg? FACtory *STB?
Arguments	<NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if *PSC is 1. If *PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.
Examples	*SRE 48 sets the bits in the SRER to binary 00110000. *SRE? might return 32, showing that the bits in the SRER have the binary value of 00100000.

***STB? (Query Only)**

The *STB? (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to Registers.

Group	Status and Error
Syntax	*STB?

Related Commands *CLS
 DESE
 *ESE
 *ESR?
 EVENT?
 EVMsg?
 FACtory
 *SRE

Returns <NR1>

Examples *STB? might return 96, showing that the SBR contains the binary value 01100000.

TEKSecure (No Query Form)

This command initializes, for the current user, both waveform and setup memories, overwriting any previously stored data.

Equivalent to invoking Teksecure from the Utility menu. This is a time-consuming operation (3 to 5 minutes) and the instrument is inoperable until the TekSecure operation is complete.

Group Miscellaneous

Syntax TEKSecure

Examples TEKSECURE initializes both waveform and setup memories.

TIME

This command sets the time in the form *hh:mm:ss* where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

Conditions Time can only be changed in increments of 1 hour maximum.

There is a 24-hour total time change limit. Time changes cannot cross the 00:00:00 boundary.

Group	Miscellaneous
Syntax	TIME <QString> TIME?
Related Commands	DATE?
Arguments	<QString> is a quoted string representing the desired time.
Examples	TIME "13:02:10" sets the time to 1:02 p.m. and 10 seconds. TIME? might return :TIME "14:05:17", indicating the current time is set to 2:05 p.m. and 17 seconds.

TIME:ZONE

This command sets the time zone to the one specified.

Group	Miscellaneous
Syntax	TIME:ZONE <QString> TIME:ZONE?
Arguments	<QString> is a quoted string representing the desired time zone.
Examples	TIME:ZONE "America/Yellowknife" sets the time zone to UTC delta -7. TIME:ZONE? might return :TIME:ZONE "America/Los_Angeles".

TIME:ZONE:UTCDELTA

This command sets or queries the time zone using the difference between the desired time zone and UTC.

Group	Miscellaneous
Syntax	TIME:ZONE:UTCDELTA <NR3> TIME:ZONE:UTCDELTA?

Arguments <NR3> is the specified number of hours difference between the desired time zone and UTC which is equivalent to GMT. The deltas supported are: -12.00, -11.00, -10.00, -9.30, -9.00, -8.30, -8.00, -7.00, -6.00, -5.00, -4.00, -3.30, -3.00, -2.00, -1.00, 0.0, 1.00, 2.00, 3.00, 3.30, 4.00, 4.30, 5.00, 5.30, 6.00, 6.30, 7.00, 8.00, 9.00, 9.30, 10.00, 10.30, 11.00, 11.30, 12.00

Examples TIME:ZONE:UTCDELTA -7.0e0 sets the time zone to America/Yellowknife.
TIME:ZONE:UTCDELTA? might return :TIME:ZONE:UTCDELTA -8.0000.

TOTALuptime? (Query Only)

Total number of hours the instrument has been turned on since the NV memory was last programmed, usually during the initial manufacturing process.

Group Miscellaneous

Syntax TOTALuptime?

Returns The total number of hours the instrument has been turned on since the NV memory was last programmed.

Examples TOTALUPTIME? might return :TOTALUPTIME 756 indicating the up time is 756 minutes.

TOUCHSCREEN:STATE

This sets or queries the enabled state of the touch screen only. This command is equivalent to pushing the Touch Off button on the front panel.

To completely disable front panel operation, use the command LOCK ALL. To re-enable the front panel, send the command LOCK NONE.

Group Miscellaneous

Syntax TOUCHSCREEN:STATE {0|1|OFF|ON}
TOUCHSCREEN:STATE?

Related Commands [LOCK](#)

Arguments 0 disables the touch screen.
 ON enables the touch screen.
 OFF disables the touch screen.

Examples TOUCHSCREEN:STATE OFF disables the touch screen.
 TOUCHSCREEN:STATE? might return :TOUCHSCREEN:STATE 1 indicating the touch screen is enabled.

*TRG (No Query Form)

Performs a group execute trigger on commands defined by *DDT.

Group Miscellaneous

Syntax *TRG

Related Commands [*DDT](#)

Examples *TRG immediately executes all commands that have been defined by *DDT.

TRIGger

This command forces a trigger event to occur. The query returns the current trigger parameters for the instrument.

Group Trigger

Syntax TRIGger FORCE
 TRIGger?

Arguments FORCE creates a trigger event. If TRIGger:STATE is set to READY, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to pressing the Force button on the front panel.

Examples TRIGGER FORCE forces a trigger event to occur.
 TRIGGER? returns the current trigger parameters for the instrument.

TRIGger:A

This command sets the A trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

Group	Trigger
Syntax	TRIGger:A SETLevel TRIGger:A?
Arguments	SETLevel sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.
Examples	TRIGger:A SETLEVEL sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal. TRIGger:A? returns current trigger parameters.

TRIGger:A:BUS:B<x>:CAN:CONDition

This command sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:CONDition {SOF FRAMeType IDentifier DATA IDANDDATA EOF ERROR FDBITS} TRIGger:A:BUS:B<x>:CAN:CONDition?
Arguments	SOF enables triggering on the start of frame. FDBITS enables triggering on the values of the BRS and ESI bits in an FD packet. FRAMeType enables triggering on the type of frame. IDentifier enables triggering on a matching identifier.

DATA enables triggering on matching data.

IDANDDATA enables triggering on a matching identifier and matching data.

EOF enables triggering on the end of frame.

ERROR enables triggering on a specified error condition.

Examples TRIGGER:A:BUS:B1:CAN:CONDITION? might return
:TRIGGER:A:BUS:B1:CAN:CONDITION EOF indicating an end of file
condition.

TRIGGER:A:BUS:B1:CAN:CONDITION DATA enables triggering on matching
CAN data.

TRIGger:A:BUS:B<x>:CAN:DATA:DIRection

This command sets the data direction (read, write or “nocare”) to be used to search on a CAN bus signal. The trigger condition must be set to IDentifier (using [TRIGger:A:BUS:B<x>:CAN:CONDition](#)). The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:CAN:DATA:DIRection {READ|WRITE|NOCARE}
TRIGger:A:BUS:B<x>:CAN:DATA:DIRection?

Arguments READ sets the CAN data direction to READ.
WRITE sets the CAN data direction to WRITE.
NOCARE sets the CAN data direction to either.

Examples TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION WRITE sets the CAN data
direction to Write.
TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION? might return
:TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION NOCARE indicating the data
direction can be either read or write.

TRIGger:A:BUS:B<x>:CAN:DATA:OFFSet

This command sets or queries the data offset value, in bytes, to use when triggering on the CAN data field. The bus number is specified by x. The trigger condition must be set to DATA or IDANDDATA.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:DATA:OFFSet <NR1> TRIGger:A:BUS:B<x>:CAN:DATA:OFFSet?
Related Commands	BUS:B<x>:CAN:STANDARD TRIGger:A:BUS:B<x>:CAN:DATA:SIZE TRIGger:A:BUS:B<x>:CAN:DATA:VALUE
Arguments	<p><NR1> is an integer whose minimum and default values are -1 (don't care), and the maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD).</p> <p>The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.</p>
Examples	<p>TRIGGER:A:BUS:B1:CAN:DATA:OFFSET 5 sets the CAN data offset to 5 bytes.</p> <p>TRIGGER:A:BUS:B1:CAN:DATA:OFFSET? might return 7, indicating the CAN data offset is 7 bytes.</p> <p>If the CAN standard is set for CAN 2.0, and the trigger data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5).</p> <p>If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD, and the trigger data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).</p>

TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier

This command sets the qualifier (<, >, =, ≠, ≤, ≥) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA

(using `TRIGger:A:BUS:B<x>:CAN:CONDition`). The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	<code>TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier</code> <code>{LESSthan MOREthan Equal UNEQual LESSEQual MOREEQual}</code> <code>TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier?</code>
Arguments	<p><code>LESSthan</code> sets the instrument to trigger when the data is less than the qualifier value.</p> <p><code>MOREthan</code> sets the instrument to trigger when the data is greater than the qualifier value.</p> <p><code>Equal</code> sets the instrument to trigger when the data is equal to the qualifier value.</p> <p><code>UNEQual</code> sets the instrument to trigger when the data is not equal to the qualifier value.</p> <p><code>LESSEQual</code> sets the instrument to trigger when the data is less than or equal to the qualifier value.</p> <p><code>MOREEQual</code> sets the instrument to trigger when the data is greater than or equal to the qualifier value.</p>
Examples	<p><code>TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN</code> sets the instrument to trigger when the data is less than the qualifier value.</p> <p><code>TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER?</code> might return <code>:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN</code>, indicating that the instrument is set to trigger when the data is less than the qualifier value.</p>

`TRIGger:A:BUS:B<x>:CAN:DATA:SIZE`

This command sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal. The trigger condition must be set to `IDANDDATA` or `DATA` (using `TRIGger:A:BUS:B<x>:CAN:CONDition`). The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
-------------------	---

Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:DATA:SIZE <NR1> TRIGger:A:BUS:B<x>:CAN:DATA:SIZE?
Arguments	<NR1> is the length of the data string in bytes.
Examples	TRIGger:A:BUS:B1:CAN:DATA:SIZE 2 sets the data size to 2 bytes. TRIGger:A:BUS:B1:CAN:DATA:SIZE? might return :TRIGGER:A:BUS:B1:CAN:DATA:SIZE 1 indicating the data size is set to 1 byte.

TRIGger:A:BUS:B<x>:CAN:DATA:VALue

This command sets the binary data value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using [TRIGger:A:BUS:B<x>:CAN:CONDition](#)). The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:DATA:VALue <QString> TRIGger:A:BUS:B<x>:CAN:DATA:VALue?
Arguments	<QString> is the data value in binary format. The only allowed characters in the QString are 0, 1, and X.
Examples	TRIGGER:A:BUS:B1:CAN:DATA:VALUE "1011" sets the CAN data value to 1011. TRIGGER:A:BUS:B1:CAN:DATA:VALUE? might return :TRIGGER:A:BUS:B1:CAN:DATA:VALUE "XXXXXXXX" indicating the data valule is set to XXXXXXXX.

TRIGger:A:BUS:B<x>:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:ERRType {ACKMISS BITSTUFFing FORMERror ANYERror} TRIGger:A:BUS:B<x>:CAN:ERRType?
Arguments	ACKMISS specifies triggering on a missing ACK field. BITSTUFFing specifies triggering on a bit stuffing error. FORMERror specifies triggering on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO. ANYERror specifies triggering on any error type.
Examples	TRIGGER:A:BUS:B1:CAN:ERRTYPE ACKMISS specifies triggering on any missing ACK fields. TRIGGER:A:BUS:B1:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is triggering on all error types.

TRIGger:A:BUS:B<x>:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:FD:BRSBit {ONE ZERO NOCARE} TRIGger:A:BUS:B<x>:CAN:FD:BRSBit?

Arguments	<p>ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).</p> <p>ZERO filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).</p> <p>NOCARE disables filtering of CAN FD packets on the BRS bit.</p>
Examples	<p>TRIGGER:A:BUS:B1:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1.</p> <p>TRIGGER:A:BUS:B1:CAN:FD:BRSBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the BRS bit value.</p>

TRIGger:A:BUS:B<x>:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:FD:ESIBit {ONE ZERO NOCARE} TRIGger:A:BUS:B<x>:CAN:FD:ESIBit?
Arguments	<p>ONE filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).</p> <p>ZERO filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).</p> <p>NOCARE disables filtering of CAN FD packets on the ESI bit.</p>
Examples	<p>TRIGGER:A:BUS:B1:CAN:FD:ESIBit ONE specifies filtering CAN FD packets for those where the ESI bit has a value of 1.</p> <p>TRIGGER:A:BUS:B1:CAN:FD:ESIBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.</p>

TRIGger:A:BUS:B<x>:CAN:FRAMeType

This command sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal. The trigger condition must be set to FRAMeType (using [TRIGger:A:BUS:B<x>:CAN:CONDition](#)). B<x> is the bus number. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:FRAMeType {DATA REMOte ERRor OVERLoad} TRIGger:A:BUS:B<x>:CAN:FRAMeType?
Arguments	DATA specifies a data frame type. REMOte specifies a remote frame type. ERRor specifies an error frame type. OVERLoad specifies an overload frame type.
Examples	TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA sets the CAN trigger frame type to DATA. TRIGGER:A:BUS:B1:CAN:FRAMETYPE? might return :TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA indicating the frame type is data.

TRIGger:A:BUS:B<x>:CAN:IDentifier:MODe

This command sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using [TRIGger:A:BUS:B<x>:CAN:CONDition](#)). The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:CAN:IDentifier:MODe {STandard EXTended} TRIGger:A:BUS:B<x>:CAN:IDentifier:MODe?

Arguments	STANDARD specifies the standard addressing mode. EXTENDED specifies the extended addressing mode.
Examples	TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE EXTENDED sets the addressing mode to extended. TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE? might return :TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE STANDARD indicating the address mode is standard.

TRIGGER:A:BUS:B<x>:CAN:IDENTIFIER:VALUE

This command sets the binary address value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATA (using [TRIGGER:A:BUS:B<x>:CAN:CONDITION](#)). The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGGER:A:BUS:B<x>:CAN:IDENTIFIER:VALUE <QString> TRIGGER:A:BUS:B<x>:CAN:IDENTIFIER:VALUE?
Arguments	<QString> is up to 29 bits specifying the binary identifier value. The only allowed characters in the QString are 0, 1, and X.
Examples	TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "1011" sets the identifier value to 1011. TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE? might return :TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "XXXXXXXXXX" indicating the identifier values is XXXXXXXXXXXX.

TRIGGER:A:BUS:B<x>:I2C:ADDRESS:MODE

This command specifies the I2C address mode to 7 or 10-bit. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
-------------------	---

Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:I2C:ADDRESS:MODE {ADDR7 ADDR10} TRIGger:A:BUS:B<x>:I2C:ADDRESS:MODE?
Arguments	ADDR7 specifies the 7-bit I ² C address mode. ADDR10 specifies the 10-bit I ² C address mode.
Examples	TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR10 sets the I2C address mode to 10-bit. TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE? might return :TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR7 indicating the address mode is set to the 7-bit mode.

TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALue

This command specifies the binary address string used for the I²C trigger if the trigger condition is ADDRESS or ADDRANDDATA. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALue <QString> TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALue?
Arguments	<QString> is up to 7 or 10-bits depending on the address mode that specifies the address. The only allowed characters in the QString are 0, 1, and X.
Examples	TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "1011" sets the I ² C address value to XXX1011. TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE? might return :TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "XXXXXXX" indicating the address value is set to XXXXXXXX.

TRIGger:A:BUS:B<x>:I2C:CONDition

This command specifies the trigger condition for an I²C trigger. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:I2C:CONDition
{START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}

TRIGger:A:BUS:B<x>:I2C:CONDition?

Arguments START specifies a search based on start condition.
STOP specifies a search based on stop condition.
REPEATstart specifies a search based on repeat of start condition.
ACKMISS specifies a search based on missing acknowledgement condition.
ADDRESS specifies a search based on address.
DATA specifies a search based on data.
ADDRANDDATA specifies a search based on address and data.

Examples TRIGGER:A:BUS:B1:I2C:CONDITION START specifies start as the I²C trigger condition.

TRIGGER:A:BUS:B1:I2C:CONDITION? might return
:TRIGGER:A:BUS:B1:I2C:CONDITION START indicating the condition is set to the start condition.

TRIGger:A:BUS:B<x>:I2C:DATA:DIRection

This command specifies the I²C trigger type to be valid on a Read, Write, or Either condition. Read or write is indicated by the R/W bit in the I²C protocol. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax	TRIGger:A:BUS:B<x>:I2C:DATA:DIRection {READ WRITE NOCARE} TRIGger:A:BUS:B<x>:I2C:DATA:DIRection?
Arguments	READ specifies read as the data direction. WRITE specifies write as the data direction. NOCARE specifies either as the data direction.
Examples	TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION WRITE specifies write as the I2C data direction. TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION? might return :TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION NOCARE indicating the data direction is either read or write.

TRIGger:A:BUS:B<x>:I2C:DATA:SIZE

This command specifies the length of the data string in bytes to be used for an I2C trigger if the trigger condition is DATA or ADDRANDDATA. Applies to bus <x>, where the bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:I2C:DATA:SIZE <NR1> TRIGger:A:BUS:B<x>:I2C:DATA:SIZE?
Arguments	<NR1> is the length of the data string in bytes.
Examples	TRIGger:A:BUS:B1:I2C:DATA:SIZE 1 sets the data size to 1 byte. TRIGger:A:BUS:B1:I2C:DATA:SIZE? might return :TRIGGER:A:BUS:B1:I2C:DATA:SIZE 1 indicating the size is set to 1 byte.

TRIGger:A:BUS:B<x>:I2C:DATA:VALue

This command specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:I2C:DATA:VALue <QString> TRIGger:A:BUS:B<x>:I2C:DATA:VALue?
Arguments	<QString> is the binary data string, where the number of bits is 8 times the number of bytes specified. The only allowed characters in the string are 0, 1, and X.
Examples	TRIGger:A:BUS:B1:I2C:DATA:VALue "11001101" sets the data value to 1100101. TRIGger:A:BUS:B1:I2C:DATA:VALue? might return :TRIGGER:A:BUS:B1:I2C:DATA:VALUE "XXXXXXXX" indicating the data value is XXXXXXXX.

TRIGger:A:BUS:B<x>:LIN:CONDition

This command specifies the trigger condition for LIN. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:LIN:CONDition {SYNCfield IDenti fier DATA IDANDDATA WAKEup SLEEP ERRor} TRIGger:A:BUS:B<x>:LIN:CONDition?
Arguments	SYNCfield sets the LIN trigger condition to sync field. IDenti fier sets the LIN trigger condition to identifier. DATA sets the LIN trigger condition to data. IDANDDATA sets the LIN trigger condition to id and data. WAKEup sets the LIN trigger condition to wake up. SLEEP sets the LIN trigger condition to sleep.

ERROR sets the LIN trigger condition to error.

Examples TRIGGER:A:BUS:B1:LIN:CONDITION ERROR sets the LIN trigger condition to error.

TRIGGER:A:BUS:B1:LIN:CONDITION? might return
 TRIGGER:A:BUS:B1:LIN:CONDITION SYNCFIELD indicating the LIN trigger condition is sync field.

TRIGger:A:BUS:B<x>:LIN:DATA:HIVALue

This command specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OUTRANGE. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:LIN:DATA:HIVALue <QString>
 TRIGger:A:BUS:B<x>:LIN:DATA:HIVALue?

Arguments <QString> is a quoted string that is the binary data string used for LIN trigger if the trigger condition is ID or IDANDDATA.

Examples TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "11001010" sets the high value to 11001010.

TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE? might return
 TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "XXXXXXXX" indicating the high value is don't care.

TRIGger:A:BUS:B<x>:LIN:DATA:QUALifier

This command specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:LIN:DATA:QUALifier
 {LESSthan|MOREthan|EQua1|UNEQua1|LESSEQua1|MOREEQua1|
 INrange|OUTrange}

TRIGger:A:BUS:B<x>:LIN:DATA:QUALifier?

Arguments LESSthan sets the LIN data qualifier to less than.
 MOREthan sets the LIN data qualifier to greater than.
 EQua1 sets the LIN data qualifier to equal.
 UNEQua1 sets the LIN data qualifier to not equal.
 LESSEQua1 sets the LIN data qualifier to less than or equal.
 MOREEQua1 sets the LIN data qualifier to greater than or equal.
 INrange sets the LIN data qualifier to in range.
 OUTrange sets the LIN data qualifier to out of range.

Examples TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER OUTRANGE sets the data qualifier to out of range.
 TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER? might return
 TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER EQUAL indicating the data qualifier is set to equal.

TRIGger:A:BUS:B<x>:LIN:DATA:SIZE

This command specifies the length of the data string in bytes to be used for LIN trigger. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:LIN:DATA:SIZE <NR1>
 TRIGger:A:BUS:B<x>:LIN:DATA:SIZE?

Arguments <NR1> is the size of the data string in bytes.

Examples TRIGGER:A:BUS:B1:LIN:DATA:SIZE 8 sets the data size to 8 bytes.
 TRIGGER:A:BUS:B1:LIN:DATA:SIZE? might return
 TRIGGER:A:BUS:B1:LIN:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:A:BUS:B<x>:LIN:DATA:VALue

This command specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:LIN:DATA:VALue <QString>
 TRIGger:A:BUS:B<x>:LIN:DATA:VALue?

Arguments <QString> is a quoted string that is the LIN trigger data value.

Examples TRIGGER:A:BUS:B1:LIN:DATA:VALUE "11001101" sets the data value to 11001101.
 TRIGGER:A:BUS:B1:LIN:DATA:VALUE? might return
 TRIGGER:A:BUS:B1:LIN:DATA:VALUE "XXXXXXXX" indicating the data value is don't care.

TRIGger:A:BUS:B<x>:LIN:ERRTYPE

This command specifies the error type be used for LIN trigger. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:LIN:ERRTYPE {SYNC|PARity|Checksum}
 TRIGger:A:BUS:B<x>:LIN:ERRTYPE?

Arguments	<p>SYNC sets the LIN error type to SYNC.</p> <p>PARi ty sets the LIN error type to parity.</p> <p>CHECKsum sets the LIN error type to checksum.</p>
Examples	<p>TRIGGER:A:BUS:B1:LIN:ERRTYPE CHECKSUM sets the LIN error type to checksum.</p> <p>TRIGGER:A:BUS:B1:LIN:ERRTYPE? might return</p> <p>TRIGGER:A:BUS:B1:LIN:ERRTYPE SYNC indicating the LIN error type is SYNC.</p>

TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue

This command specifies the binary address string used for LIN bus trigger if the trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue <QString> TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue?
Arguments	<QString> is the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.
Examples	<p>TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "110010" sets the identifier value to 110010.</p> <p>TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE? might return</p> <p>TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "XXXXXX" indicating the identifier value is XXXXXX.</p>

TRIGger:A:BUS:B<x>:PARAllel:DATA:VALue

This command specifies the binary data string used for a Parallel Bus trigger. The bus number is specified by x.

Group	Trigger
--------------	---------

Syntax	TRIGger:A:BUS:B<x>:PARAllel:DATA:VALue <QString> TRIGger:A:BUS:B<x>:PARAllel:DATA:VALue?
Arguments	<QString> is the binary data string used for a Parallel Bus trigger.
Examples	TRIGger:A:BUS:B1:PARAllel:DATA:VALue "11001101" sets the data value to 11001101. TRIGger:A:BUS:B1:PARAllel:DATA:VALue? might return :TRIGGER:A:BUS:B1:PARALLEL:DATA:VALUE "X" indicating the data value is X.

TRIGger:A:BUS:B<x>:RS232C:CONDition

This command specifies the condition for an RS-232C trigger, where the bus number is specified by < >x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:RS232C:CONDition {START EOP DATA PARItYerror} TRIGger:A:BUS:B<x>:RS232C:CONDition?
Arguments	START sets the Trigger on condition to Start. EOP sets the Trigger on condition to End of Packet. DATA sets the Trigger on condition to Data. PARItYerror sets the Trigger on condition to Parity Error.
Examples	TRIGger:A:BUS:B1:RS232C:CONDition DATA sets the trigger on condition to data. TRIGger:A:BUS:B1:RS232C:CONDition? might return :TRIGGER:A:BUS:B1:RS232C:CONDITION START indicating start is the trigger condition.

TRIGger:A:BUS:B<x>:RS232C:DATA:SIZE

This command sets or queries the length of the data string in bytes to be used for an RS-232C trigger when the trigger condition is Data. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:RS232C:DATA:SIZE <NR3> TRIGger:A:BUS:B<x>:RS232C:DATA:SIZE?
Arguments	<NR3> specifies the data size in bytes.
Examples	TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 3 sets three bytes data size for the RS-232C bus B1 trigger. TRIGGER:A:BUS:B1:RS232C:DATA:SIZE? might return :TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 2, indicating that the data size for the RS-232C bus B1 trigger is set to two bytes.

TRIGger:A:BUS:B<x>:RS232C:DATA:VALue

This command sets or queries the data address string used for the RS-232 bus trigger when the trigger condition is set to Data. The bus number is specified by x.

Conditions	Requires option 2-SERIAL or 2-ULTIMATE.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:RS232C:DATA:VALue <QString> TRIGger:A:BUS:B<x>:RS232C:DATA:VALue?
Arguments	<QString> specifies the address value. The argument is a string of 0, 1, or X representing a binary number.
Examples	TRIGGER:A:BUS:B1:RS232C:DATA:VALUE "011XX11" sets the data address string used for the RS-232 bus trigger to 011XX11.

TRIGGER:A:BUS:B1:RS232C:DATA:VALUE? might return
:TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXX01", indicating
that the data address string used for the RS-232 bus trigger is set to "XXXXXX01"

TRIGger:A:BUS:B<x>:SENT:CONDition

This command sets or queries the trigger condition for a SENT bus.

Conditions Requires option SRAUTOSEN.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:CONDition {START|FAST|SLOW|ERRor}
TRIGger:A:BUS:B<x>:SENT:CONDition?

Arguments B<x> is the number of the bus waveform.
START sets triggering on start of packet.
FAST sets triggering on fast channel packets.
SLOW sets triggering on slow channel packets.
ERRor sets triggering on errors.

Examples TRIGGER:A:BUS:B1:SENT:CONDITION START specifies triggering on start
of packet.
TRIGGER:A:BUS:B1:SENT:CONDITION? might return
TRIGGER:A:BUS:B1:SENT:CONDITION ERROR, indicating the bus
is triggering on errors.

TRIGger:A:BUS:B<x>:SENT:ERRType

This command sets or queries the error type to be used when triggering on SENT
data.

Conditions Requires option 5SRAUTOSEN.
The SENT bus trigger condition is set to ERRor.

Group Trigger

Syntax	TRIGger:A:BUS:B<x>:SENT:ERRType CRC TRIGger:A:BUS:B<x>:SENT:ERRType?
Related Commands	TRIGger:A:BUS:B<x>:SENT:CONDition
Arguments	B<x> is the number of the bus waveform. CRC specifies triggering on CRC errors.
Examples	TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC sets the bus to trigger on CRC errors. TRIGGER:A:BUS:B1:SENT:ERRTYPE? might return :TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC to indicate the bus is triggering on CRC errors.

TRIGger:A:BUS:B<x>:SENT:ERRType:CRc

This command sets or queries the CRC error type to be used when triggering on SENT data.

Conditions	Requires option SRAUTOSEN. The SENT bus trigger condition is set to ERRor. The ERRType is set to CRC.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:SENT:ERRType:CRc {FAST SLOW} TRIGger:A:BUS:B<x>:SENT:ERRType:CRc?
Related Commands	TRIGger:A:BUS:B<x>:SENT:CONDition TRIGger:A:BUS:B<x>:SENT:ERRType
Arguments	B<x> is the number of the bus waveform. FAST specifies triggering on CRC errors in only the Fast Channel. SLOW specifies triggering on CRC errors in only the slow channel.
Examples	TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRc FAST sets the bus to trigger on Fast Channel CRC errors.

TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC? might return
 TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC FAST to indicate that the
 bus is triggering on Fast Channel CRC errors.

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue <QString>
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

Arguments Bus<x> is the bus identifier number.

<Qstring> sets the Fast Channel 1 binary data high value.

Examples TRIGger:A:BUS:B12:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXXX"
 sets the SENT bus B12 Fast Channel 1 high value to XXXXXXXXXXXXXXXX, or
 "don't care."

TRIGger:A:BUS:B3:SENT:FAST:CHAN1A:HIVALue? might return
 TRIGGER:A:BUS:B3:SENT:FAST:CHAN1A:HIVALUE "0101XXXX1111",
 indicating the SENT bus B3 Fast Channel 1 high is set to the binary value
 0101XXXX1111.

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT
 fast packet bus data for device channel 1.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group	Trigger
Syntax	<pre>TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A: QUALifier {EQual UNEQual LESSthan MORethan LESSEQual MOREEQual INrange OUTrange} TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier?</pre>
Related Commands	<p>TRIGger:A:BUS:B<x>:SENT:CONDition</p> <p>TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue</p>
Arguments	<p>B<x> is the number of the bus.</p> <p>EQual sets the qualifier as Equal.</p> <p>LESSEQual sets the qualifier as Less Than or Equal to.</p> <p>LESSThan sets the qualifier as Less Than.</p> <p>MOREEQual sets the qualifier as More Than or Equal to.</p> <p>MOREThan sets the qualifier as More Than.</p> <p>UNEQual sets the qualifier as Unequal.</p> <p>INrange sets the qualifier to inside a range.</p> <p>OUTrange sets the qualifier to outside a range.</p>
Examples	<p>TRIGGER:A:BUS:B2:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL sets the fast channel 1 data qualifier trigger to not equal for bus 2.</p> <p>TRIGGER:A:BUS:B5:SENT:FAST:CHAN1A:QUALIFIER? might return TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier trigger is set to greater than or equal for bus 5.</p>

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.

Conditions	<p>Requires option SRAUTOSEN.</p> <p>The search trigger condition must be set to FAST.</p>
Group	Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue <Qstring>
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)
[TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier](#)

Arguments B<x> is the number of the bus waveform.
 <Qstring> is the Fast Channel 1 value on which to trigger.

Examples TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the Fast Channel 1 value to trigger on XXXXXXXXXXXXX, or "don't care."
 TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "0000XXXX1111" to indicate the Fast Channel 1 binary trigger value is 0000XXXX1111.

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
 The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue <Qstring>
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

Arguments Bus<x> is the bus identifier number.
 <Qstring> sets the Fast Channel 2 high binary data value.

Examples TRIGger:A:BUS:B2:SENT:FAST:CHAN2B:HIVALue "10000000000" sets the SENT bus B2 Fast Channel 2 high value to 10000000000.

TRIGger:A:BUS:B4:SENT:FAST:CHAN2B:HIVALue? might return
 TRIGGER:A:BUS:B4:SENT:FAST:CHAN2B:HIVALUE "0101XXXX1111",
 indicating the SENT bus B4 Fast Channel 2 high is set to the binary value
 0101XXXX1111.

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 2.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
 QUALifier {Equal|UNEQUAL|LESSThan|MOREthan|
 LESSEQUAL|MOREEQUAL|INrange|OUTrange}
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

[TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:VALue](#)

Arguments B<x> is the number of the bus.

EQUAL sets the qualifier as Equal.

LESSEQUAL sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQUAL sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

UNEQUAL sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples TRIGGER:A:BUS:B2:SENT:FAST:CHAN2B:QUALIFIER UNEQUAL sets the fast
 channel 2 data qualifier trigger to not equal for bus 2.

TRIGGER:A:BUS:B5:SENT:FAST:CHAN2B:QUALIFIER? might return
 TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:QUALIFIER MOREEQUAL to
 indicate that the fast channel 2 data qualifier trigger is set to greater than or equal
 for bus 5.

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when triggering on a SENT bus signal. The trigger condition must be set to FAST.

Conditions Requires option SRAUTOSEN.

The search condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:VALue <Qstring>
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:VALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

[TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier](#)

Arguments B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 2 binary value on which to trigger.

Examples TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "111111111111" sets the
 Fast Channel 2 value on which to trigger to 111111111111.

TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE? might return
 :TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "000000000000" to
 indicate the Fast Channel 2 binary value on which to trigger is 000000000000.

TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:HIVALue

This command sets or queries the high binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:HIVALue <QString>
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:HIVALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)
[BUS:B<x>:SENT:NUMCHANnel](#)
[BUS:B<x>:SENT:NIBBLECount](#)

Arguments Bus<x> is the bus identifier number.
<Qstring> sets the Fast Channel 1 counter binary value.

Examples TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:HIVALUE "XXXXXXXX" sets the Fast Channel 1 secure counter high value to "don't care."
TRIGGER:A:BUS:B3:SENT:FAST:COUNTER:HIVALUE? might return TRIGGER:A:BUS:B3:SENT:FAST:COUNTER:HIVALUE "11110000" to indicate a binary value of 11110000.

TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for the secure format counter.

Conditions Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:QUALifier {EQual|UNEQual|LESSthan|MORethan|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:QUALifier?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

[TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:VALue](#)

Arguments B<x> is the number of the bus.
 EQUa1 sets the qualifier as Equal.
 LESSEQUa1 sets the qualifier as Less Than or Equal to.
 LESSThan sets the qualifier as Less Than.
 MOREEQUa1 sets the qualifier as More Than or Equal to.
 MORETHan sets the qualifier as More Than.
 UNEQUa1 sets the qualifier as Unequal.
 INrange sets the qualifier to inside a range.
 OUTrange sets the qualifier to outside a range.

Examples TRIGGER:A:BUS:B2:SENT:FAST:COUNTER:QUALIFIER UNEQUAL sets the fast channel 2 counter qualifier trigger to not equal for bus 2.
 TRIGGER:A:BUS:B5:SENT:FAST:COUNTER:QUALIFIER? might return TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:QUALIFIER MOREEQUAL to indicate that the fast channel 2 counter qualifier trigger is set to greater than or equal for bus 5.

TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:VALue

This command sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
 The search condition must be set to FAST.
 The number of channels must be set to 1.
 The nibble count must be set to 6.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:VALue <Qstring>
 TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:VALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

[BUS:B<x>:SENT:NIBBLECount](#)

[BUS:B<x>:SENT:NUMCHANnel](#)

Arguments B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 fast message counter binary value on which to trigger.

Examples TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 fast message counter value to "don't care."

TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE "00001111" to indicate the Fast Channel 1 fast message counter binary value on which to trigger is 00001111.

TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue <Qstring>
TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)
[BUS:B<x>:SENT:NIBBLECount](#)
[BUS:B<x>:SENT:NUMCHANnel](#)

Arguments B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 inverted nibble binary value on which to trigger.

Examples TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX" sets the Fast Channel 1 secure inverted nibble value to "don't care."

TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "1111" to indicate a binary value of 1111.

TRIGger:A:BUS:B<x>:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.
The search trigger condition must be set to FAST.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:FAST:STATus:VALue <Qstring>
TRIGger:A:BUS:B<x>:SENT:FAST:STATus:VALue?

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)

Arguments B<x> is the number of the bus waveform.
<Qstring> is the binary status value on which to trigger.

Examples TRIGger:A:BUS:B1:SENT:FAST:STATus:VALue "XXXX" sets the binary status trigger value to XXXX.

TRIGGER:A:BUS:B1:SENT:FAST:STATus:VALue? might return :TRIGGER:A:BUS:B1:SENT:FAST:STATus:VALue "1010" to indicate a binary trigger value is 1010.

TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT pause pulses.

Conditions Requires option SRAUTOSEN.
The SENT bus trigger condition must be set to PAUSE.

Group	Trigger
Syntax	<pre>TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier {EQUal UNEQUal LESSThan MOREthan LESSEQUal MOREEQUal Inrange OUTrange} TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier?</pre>
Related Commands	<p>TRIGger:A:BUS:B<x>:SENT:CONDition</p> <p>SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue</p> <p>SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue</p>
Arguments	<p>B<x> is the bus identifier number.</p> <p>EQUal sets the qualifier as Equal.</p> <p>INrange sets the qualifier to in range.</p> <p>LESSEQUal sets the qualifier as Less Than or Equal to.</p> <p>LESSThan sets the qualifier as Less Than.</p> <p>MOREEQUal sets the qualifier as More Than or Equal to.</p> <p>MOREThan sets the qualifier as More Than.</p> <p>OUTrange sets the qualifier to out of range.</p> <p>UNEQUal sets the qualifier as Unequal.</p>
Examples	<pre>TRIGGER:A:BUS:B2:SENT:PAUSE:QUALifier UNEQUAL sets the pause pulse qualifier trigger to not equal for bus 2.</pre> <pre>TRIGGER:A:BUS:B1:SENT:PAUSE:QUALIFIER? might return :TRIGGER:A:BUS:B1:SENT:PAUSE:QUALIFIER INRANGE to indicate that the pause pulse qualifier is set to trigger inside the range of ticks low and ticks high on bus 1.</pre>

TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when triggering on a SENT bus signal.

Conditions	<p>Requires option SRAUTOSEN.</p> <p>The SENT bus trigger condition must be set to SLOW.</p>
-------------------	--

Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:HIVALue <Qstring> TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:HIVALue?
Related Commands	TRIGger:A:BUS:B<x>:SENT:CONDition
Arguments	BUS<x> is the bus identifier number. <Qstring> sets the binary Slow channel data value.
Examples	TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE "XXXXXXXX" sets the Slow data high value to "don't care." TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE? might return TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE "11110001", to indicate triggering on Slow data values that match 11110001.

TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:QUALifier

This command sets or queries the binary identifier value to use when triggering on a SENT bus signal.

Conditions	Requires option SRAUTOSEN The SENT bus trigger condition must be set to SLOW.
Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:QUALifier {Equal UNEQual LESSthan MOREthan LESSEQual MOREEQual} TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:QUALifier?
Related Commands	TRIGger:A:BUS:B<x>:SENT:CONDition
Arguments	B<x> is the bus identifier number. EQUal specifies the qualifier as Equal. UNEQUal specifies the qualifier as Not Equal to. LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

LESSEQUAL specifies the qualifier as Less Than or Equal to.

MOREEQUAL specifies the qualifier as More Than or Equal to.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples TRIGGER:A:BUS:B2:SENT:SLOW:DATA:QUALIFIER LESSEQUAL sets the slow channel data qualifier to less than or equal for bus 2.

TRIGGER:A:BUS:B5:SENT:SLOW:DATA:QUALIFIER? might return :TRIGGER:A:BUS:B5:SENT:SLOW:DATA:QUALIFIER EQUAL to indicate that the slow channel data qualifier is set to equal on bus 5.

TRIGGER:A:BUS:B<x>:SENT:SLOW:DATA:VALUE

This command sets or queries the binary slow channel value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN
The SENT bus trigger condition must be set to SLOW.

Group Trigger

Syntax TRIGGER:A:BUS:B<x>:SENT:SLOW:DATA:VALUE <Qstring>
TRIGGER:A:BUS:B<x>:SENT:SLOW:DATA:VALUE?

Related Commands [TRIGGER:A:BUS:B<x>:SENT:CONDITION](#)

Arguments B<x> is the bus identifier number.
<Qstring> is the binary slow channel data value.

Examples TRIGGER:A:BUS:B2:SENT:SLOW:DATA:VALUE "XXXXXXXX" sets the slow data value to "don't care" for bus 2.

TRIGGER:A:BUS:B4:SENT:SLOW:DATA:VALUE? might return :TRIGGER:A:BUS:B4:SENT:SLOW:DATA:VALUE "XXXX1111", to indicate triggering on slow data values that match XXXX1111 on bus 4.

TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:VALue

This command sets or queries the qualifier to use when triggering on SENT slow packet bus data.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:VALue <Qstring>
TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:VALue

Related Commands [TRIGger:A:BUS:B<x>:SENT:CONDition](#)
[TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:VALue](#)

Arguments B<x> is the bus identifier number.
<Qstring> is the binary identifier value.

Examples TRIGGER:A:BUS:B3:SENT:SLOW:IDENTIFIER:VALUE "XXXX" sets the identifier value to "don't care" for bus 3.
TRIGGER:A:BUS:B1:SENT:SLOW:IDENTIFIER:VALUE? might return :TRIGGER:A:BUS:B1:SENT:SLOW:IDENTIFIER:VALUE "0001" to indicate the slow channel identifier is set to 0001 on bus 1.

TRIGger:A:BUS:B<x>:SPI:CONDition

This command specifies the trigger condition for a SPI trigger. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SPI:CONDition {SS|STARTofframe|DATA}
TRIGger:A:BUS:B<x>:SPI:CONDition?

Related Commands [BUS:B<x>:SPI:IDLETime](#)
[BUS:B<x>:SPI:FRAMING](#)

Arguments B<x> is the bus identifier number.
SS specifies the Slave Selection condition.
STARTofframe is applicable when BUS:B<x>:SPI:FRAMING is set to IDLEtime. When the trigger condition is set to STARTofframe, the instrument triggers on the first SPI clock after an idle time when there are no clocks.
DATA sets the trigger condition to Master-In Slave-Out and Master-Out Slave-In.

TRIGger:A:BUS:B<x>:SPI:DATA:SIZE

This command specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATA. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group Trigger

Syntax TRIGger:A:BUS:B<x>:SPI:DATA:SIZE <NR1>
TRIGger:A:BUS:B<x>:SPI:DATA:SIZE?

Arguments B<x> is the bus identifier number.
<NR1> is the length of the data string in bytes.

Examples TRIGger:A:BUS:B1:SPI:DATA:SIZE 1 sets the data size to 1 byte.
TRIGger:A:BUS:B1:SPI:DATA:SIZE? might return
:TRIGGER:A:BUS:B1:SPI:DATA:SIZE 1 indicating the data size is
1 byte.

TRIGger:A:BUS:B<x>:SPI:DATA:VALue

This command specifies the binary data string used for SPI triggering if the trigger condition is DATA. The bus number is specified by x.

Conditions Requires option 2-SERIAL or 2-ULTIMATE.

Group	Trigger
Syntax	TRIGger:A:BUS:B<x>:SPI:DATA:VALue <QString> TRIGger:A:BUS:B<x>:SPI:DATA:VALue?
Arguments	B<x> is the bus identifier number. <QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format.
Examples	TRIGger:A:BUS:B1:SPI:DATA:VALue "11011010" sets the data value to 11011010. TRIGger:A:BUS:B1:SPI:DATA:VALue? might return :TRIGGER:A:BUS:B1:SPI:DATA:VALUE "XXXXXXXX" indicating the data value is XXXXXXXX.

TRIGger:A:BUS:SOURce

This command sets or queries the source bus for a bus trigger.

Group	Trigger
Syntax	TRIGger:A:BUS:SOURce B<x> TRIGger:A:BUS:SOURce?
Arguments	B<x> sets the selected source to the bus.
Examples	TRIGGER:A:BUS:SOURCE B2 sets the selected source for the bus trigger to Bus 2. TRIGGER:A:BUS:SOURCE? might return TRIGGER:A:BUS:SOURCE B1, indicating that the selected source for the bus trigger is set to Bus 1.

TRIGger:A:EDGE:COUPLing

This command sets or queries the type of coupling for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and choosing from the Coupling drop-down list.

Group	Trigger
--------------	---------

Syntax	TRIGger:A:EDGE:COUPling {DC HFRej LFRej NOISErej} TRIGger:A:EDGE:COUPling?
Related Commands	TRIGger:A:EDGE:SOUrce TRIGger:A:EDGE:SLOpe
Arguments	DC selects DC trigger coupling, which passes all input signals to the trigger circuitry. HFRej coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry. LFRej coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry. NOISErej coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but can require greater trigger signal amplitude.
Examples	TRIGGER:A:EDGE:COUPLING DC sets the A edge trigger coupling to DC. TRIGGER:A:EDGE:COUPLING? might return :TRIGGER:A:EDGE:COUPLING DC, indicating that the A edge trigger coupling is set to DC.

TRIGger:A:EDGE:SLOpe

This command sets or queries the slope for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and then choosing the desired Slope. This command is also equivalent to pressing the front-panel Slope button.

Group	Trigger
Syntax	TRIGger:A:EDGE:SLOpe {RISe FALL EITHer} TRIGger:A:EDGE:SLOpe?
Related Commands	TRIGger:A:EDGE:SOUrce TRIGger:A:EDGE:COUPling
Arguments	RISe specifies to trigger on the rising or positive edge of a signal. FALL specifies to trigger on the falling or negative edge of a signal.

EITHER specifies to trigger on either the rising or falling edge of a signal.

Examples TRIGGER:A:EDGE:SLOPE RISE sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

TRIGGER:A:EDGE:SLOPE? might return :TRIGGER:A:EDGE:SLOPE FALL, indicating that the A edge trigger slope is negative.

TRIGger:A:EDGE:SOUrce

This command sets or queries the source for the edge trigger. For instruments that have an Auxiliary Input, AUXiliary can be selected as trigger source.

Group Trigger

Syntax TRIGger:A:EDGE:SOUrce {CH<x>| DCH<x>_D<x>| INTerna1|
AUXiliary}
TRIGger:A:EDGE:SOUrce?

Related Commands [TRIGger:A:EDGE:SLOpe](#)
[TRIGger:A:EDGE:COUPling](#)

Arguments CH<x> specifies an analog channel as the edge trigger source.
DCH<x>_D<x> specifies a digital channel as the edge trigger source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
INTerna1 specifies a internal edge trigger source.
AUXiliary specifies the Auxiliary Input.

Examples TRIGGER:A:EDGE:SOURCE CH2 sets the A edge trigger source to input channel 2.

TRIGGER:A:EDGE:SOURCE? might return :TRIGGER:A:EDGE:SOURCE CH1, indicating that the A edge trigger source is set to input channel 1.

TRIGger:A:LEVel:CH<x>

This command sets or queries the CH<x> trigger level for an Edge, Pulse Width, Runt or Rise/Fall (Transition and Slew Rate) trigger when triggering on an analog channel waveform. Each channel can have an independent trigger level. The <x> is the channel number.

Group	Trigger
Syntax	TRIGger:A:LEVEl:CH<x> <NR3> TRIGger:A:LEVEl:CH<x>?
Arguments	<NR3> specifies the trigger level in user units (usually volts).
Examples	TRIGGER:A:LEVEL:CH1 1.5 sets the A trigger level for Channel 1 to 1.5 V. TRIGGER:A:LEVEL:CH2? might return :TRIGGER:A:LEVEL:CH2 1.3000E+00, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:A:LOGIc:DELTatime

This command specifies or queries the Logic trigger delta time value. The time value is used as part of the Logic trigger condition to determine if the duration of a logic pattern meets the specified time constraints.

Group	Trigger
Syntax	TRIGger:A:LOGIc:DELTatime <NR3> TRIGger:A:LOGIc:DELTatime?
Arguments	<NR3> the Logic trigger delta time value.
Examples	TRIGger:A:LOGIC:DELTatime 4.5e-9 sets the delta time to 4.5 ns. TRIGger:A:LOGIC:DELTatime? might return :TRIGGER:A:LOGIC:DELTATIME 4.0E-9 indicating the delta time is set to 4 ns.

TRIGger:A:LOGIc:FUNcTion

This command sets or queries the logical combination of the input channels for logic triggers. This command is equivalent to selecting Logic for the Trigger Type, and setting or viewing the Define Logic.

Group	Trigger
--------------	---------

Syntax	TRIGger:A:LOGIC:FUNCTION {AND NAND NOR OR} TRIGger:A:LOGIC:FUNCTION?
Arguments	AND specifies to trigger if all conditions are true. NAND specifies to trigger if any of the conditions are false. NOR specifies to trigger if all conditions are false. OR specifies to trigger if any of the conditions are true.
Examples	TRIGGER:A:LOGIC:FUNCTION AND sets the logical combination of channels to be true when all conditions are true. TRIGGER:A:LOGIC:FUNCTION? might return :TRIGGER:A:LOGIC:FUNCTION NAND, indicating that the instrument will trigger if the AND logic conditions are false.

TRIGger:A:LOGic:INPut:CLOCK:SOURce

This command specifies the channel to use as the clock source for logic trigger.

Group	Trigger
Syntax	TRIGger:A:LOGic:INPut:CLOCK:SOURce {CH<x> DCH<x>_D<x>} TRIGger:A:LOGic:INPut:CLOCK:SOURce?
Arguments	CH<x> specifies an analog channel as the clock source. Number of channels depends on instrument configuration. DCH<x>_D<x> specifies a digital channel as the clock source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
Examples	TRIGger:A:LOGic:INPut:CLOCK:SOURce CH3 sets the clock source to channel 3. TRIGger:A:LOGic:INPut:CLOCK:SOURce? might return TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE CH3 indicating the clock source is set to channel 3.

TRIGger:A:LOGIc:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.

Group	Trigger
Syntax	TRIGger:A:LOGIC:POLarity {POSitive NEGative EITHer} TRIGger:A:LOGIC:POLarity?
Arguments	NEGative specifies negative polarity. POSITIVE specifies positive polarity. EITHer specifies either polarity.
Examples	TRIGger:A:LOGIC:POLarity EITHER sets the polarity to either positive or negative. TRIGger:A:LOGIC:POLarity? might return :TRIGGER:A:LOGIC:POLARITY POSITIVE indicating the polarity is set to positive.

TRIGger:A:LOGIc:USECLockedge

This command specifies whether or not Logic trigger type uses clock source.

Group	Trigger
Syntax	TRIGger:A:LOGIC:USECLockedge {OFF ON 0 1} TRIGger:A:LOGIC:USECLockedge?
Arguments	ON specifies that logic trigger type uses clock source. OFF specifies that logic trigger type does not use clock source. <NR1> = 0 specifies that logic trigger type does not use clock source; any other value uses clock source.
Examples	TRIGger:A:LOGIC:USECLockedge OFF specifies that the clock edge will not be used.

TRIGger:A:LOGIC:USEClockedge? might return
:TRIGGER:A:LOGIC:USECLOCKEDGE 1 indicating that the clock edge will be used.

TRIGger:A:LOGic:WHEn

This command sets or queries the condition for generating an A or B logic trigger with respect to the defined input pattern. This command is equivalent to selecting Logic for Trigger Type, Use Clock Edge to No, and choosing a trigger condition from the Logic Pattern drop-down list.

Group	Trigger
Syntax	TRIGger:A:LOGIC:WHEn {TRUE FALSE MOREthan LESSthan EQua1 UNEQua1} TRIGger:A:LOGIC:WHEn?
Arguments	<p>TRUE triggers on an input pattern that is true.</p> <p>FALSE triggers on an input pattern that is false.</p> <p>MOREthan triggers on an input pattern that is true for a time period greater than a user defined Time Limit (DELTime) value.</p> <p>LESSthan triggers on an input pattern that is true for a time period less than a user defined Time Limit (DELTime) value.</p> <p>EQua1 triggers on an input pattern that is true for a time period equal to a user defined Time Limit (DELTime) value.</p> <p>UNEQua1 triggers on an input pattern that is true for a time period not equal to a user defined Time Limit (DELTime) value.</p>
Examples	<p>TRIGger:A:LOGIC:WHEn EQua1 specifies triggering when the input pattern is true for a time period equal to a user defined Time Limit value.</p> <p>TRIGger:A:LOGIC:WHEn? might return :TRIGGER:A:LOGIC:WHEn TRUE indicating a trigger when the logic is true.</p>

TRIGger:A:LOGICPattern:CH<x>

This command sets or queries the Logic Pattern for the specified channel. This command is used along with the Define Logic choice (LOGic:FUNcTION) to determine when the logic trigger occurs.

Group	Trigger
Syntax	TRIGger:A:LOGICPattern:CH<x> {HIGH LOW X} TRIGger:A:LOGICPattern:CH<x>?
Arguments	CH<x> specifies the channel. HIGH specifies triggering when the pattern is high. LOW specifies triggering when the pattern is low. X specifies triggering when the pattern is high or low.
Examples	TRIGger:A:LOGIC:LOGICPattern:CH2 HIGH sets the triggering logic to when pattern is high. TRIGger:A:LOGIC:LOGICPattern:CH2? might return :TRIGGER:A:LOGIC:LOGICPATTERN:CH2 X indicating the state can be high or low.

TRIGger:A:LOGICPattern:DCH<x>_D<x>

This command sets or queries the Logic Pattern for the specified digital channel. This command is used along with the Define Logic choice (LOGIC:FUNCTION) to determine when the logic trigger occurs.

Group	Trigger
Syntax	TRIGger:A:LOGICPattern:DCH<x>_D<x> {HIGH LOW X} TRIGger:A:LOGICPattern:DCH<x>_D<x>?
Arguments	DCH<x>_D<x> specifies the digital channel. The supported digital channel value is 1. The supported digital bits values are 0 to 15. HIGH specifies triggering when the pattern is high. LOW specifies triggering when the pattern is low. X specifies triggering when the pattern is high or low.
Examples	TRIGger:A:LOGIC:LOGICPattern:DCH1_D1 HIGH sets the triggering logic to when pattern is high.

TRIGger:A:LOGIC:LOGICPattern:DCH1_D1? might return
:TRIGGER:A:LOGIC:LOGICPATTERN:DCH1_D1 X indicating the state can be
high or low.

TRIGger:A:LOWerthreshold:CH<x>

This command sets or queries the A or B lower trigger level threshold for the channel, specified by x.

Group Trigger

Syntax TRIGger:A:LOWerthreshold:CH<x> <NR3>
TRIGger:A:LOWerthreshold:CH<x>?

Related Commands [TRIGger:A:UPPerthreshold:CH<x>](#)

Arguments <NR3> specifies the threshold voltage in user units.

Examples TRIGGER:A:LOWERTHRESHOLD:CH2 1.3 sets the A trigger threshold voltage for Channel 2 to 1.3 V.

TRIGGER:A:LOWERTHRESHOLD:CH3? might return
TRIGGER:A:LOWERTHRESHOLD:CH3 1.2000E+00, indicating that the A trigger threshold voltage for Channel 3 is 1.2 V.

TRIGger:A:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use [TRIGger:A:PULSEWidth:LOWLimit](#) to specify the lower value of the range.)

Group Trigger

Syntax TRIGger:A:PULSEwidth:HIGHLimit <NR3>
TRIGger:A:PULSEwidth:HIGHLimit?

Related Commands [TRIGger:A:PULSEWidth:WHEn](#)
[TRIGger:A:PULSEWidth:LOWLimit](#)

Arguments <NR3> is a floating point number that represents the higher value of the range.

Examples TRIGGER:A:PULSEwidth:HIGHLimit 110.0E-9 sets the high limit to 110.0 ns.

TRIGGER:A:PULSEwidth:HIGHLimit? might return
TRIGGER:A:PULSEwidth:HIGHLIMIT 178.88000E-9 indicates
the high limit is set to 178.88 ns.

TRIGGER:A:PULSEwidth:LOWLimit

This command specifies the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use [TRIGGER:A:PULSEwidth:HIGHLimit](#) to specify the upper limit of the range.)

This command also specifies the single limit to use, in seconds, when triggering on detection of a pulse whose duration is less than, greater than, equal to, or not equal to this time limit.

Group Trigger

Syntax TRIGGER:A:PULSEwidth:LOWLimit <NR3>
TRIGGER:A:PULSEwidth:LOWLimit?

Related Commands [TRIGGER:A:PULSEwidth:WHEn](#)
[TRIGGER:A:PULSEwidth:HIGHLimit](#)

Arguments <NR3> is a floating point number that represents the lower value of the range.

Examples TRIGGER:A:PULSEwidth:LOWLimit 100.0E-9 sets the low limit to 100.0 ns.

TRIGGER:A:PULSEwidth:HIGHLimit? might return
TRIGGER:A:PULSEwidth:LOWLIMIT 77.7600E-9 indicating the
low limit is set to 77.76 ns.

TRIGGER:A:PULSEwidth:POLarity

This command specifies the polarity for a pulse width trigger.

Group Trigger

Syntax	TRIGger:A:PULSEwidth:POLarity {NEGative POSitive} TRIGger:A:PULSEwidth:POLarity?
Arguments	NEGative specifies a negative pulse. POSitive specifies a positive pulse.
Examples	TRIGger:A:PULSEwidth:POLarity NEGATIVE sets the pulse polarity to negative. TRIGger:A:PULSEwidth:POLarity? might return TRIGGER:A:WIDTH:POLARITY POSITIVE indicating a positive pulse.

TRIGger:A:PULSEWidth:SOURce

This command specifies the source waveform for a pulse width trigger.

Group	Trigger
Syntax	TRIGger:A:PULSEwidth:SOURce {CH<x> DCH<x>_D<x>} TRIGger:A:PULSEwidth:SOURce?
Arguments	CH<x> specifies an analog input channel as the pulse-width trigger source. DCH<x>_D<x> specifies an digital input channel as the pulse-width trigger source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
Examples	TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source. TRIGGER:A:PULSEWIDTH:SOURCE? might return :TRIGGER:A:PULSEWIDTH:SOURCE CH1 indicating that channel 1 is the pulse width trigger source.

TRIGger:A:PULSEWidth:WHEn

This command specifies to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using [TRIGger:A:PULSEWidth:LOWLimit](#)), OR whose width falls outside of or within a specified range of two values (set using [TRIGger:A:PULSEWidth:LOWLimit](#) and [TRIGger:A:PULSEWidth:HIGHLimit](#)).

Group	Trigger
Syntax	<pre>TRIGger:A:PULSEwidth:WHEN {LESSthan MOREthan EQua UNEQua WITHin OUTside} TRIGger:A:PULSEwidth:WHEN?</pre>
Related Commands	<p>TRIGger:A:PULSEwidth:LOWLimit</p> <p>TRIGger:A:PULSEwidth:HIGHLimit</p> <p>TRIGger:A:PULSEwidth:SOUrce</p>
Arguments	<p>LESSthan causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:A:PULSEwidth:LOWLimit command.</p> <p>MOREthan causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:A:PULSEwidth:LOWLimit command.</p> <p>EQua causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:A:PULSEwidth:LOWLimit within a $\pm 5\%$ tolerance.</p> <p>UNEQua causes a trigger when a pulse is detected with a width greater than or less than (but not equal) the time period specified in TRIGger:A:PULSEwidth:LOWLimit within a $\pm 5\%$ tolerance.</p> <p>WITHin causes a trigger when a pulse is detected that is within a range set by two values.</p> <p>OUTside causes a trigger when a pulse is detected that is outside of a range set by two values.</p>
Examples	<pre>:TRIGger:A:PULSEwidth:WHEN LESSthan causes the instrument to trigger when a pulse is detected that is shorter than the time specified by TRIGger:A:PULSEwidth:LOWLimit.</pre> <pre>TRIGger:A:PULSEwidth:WHEN ? might return TRIGGER:A:PULSEWIDTH:WHEN GREATER THAN 2.0000E-9, indicating that a trigger is generated when a pulse is detected greater than 2 ns.</pre> <pre>TRIGger:A:PULSEwidth:WHEN? might return TRIGger:A:PULSEwidth:WHEN MOREthan indicating that a trigger is generated when a pulse is detected that is greater than the time specified by TRIGger:A:PULSEwidth:LOWLimit.</pre>

TRIGger:A:RUNT:POLarity

This command specifies the polarity for the runt trigger.

Group	Trigger
Syntax	TRIGger:A:RUNT:POLARity {EITHer NEGAtive POSitive} TRIGger:A:RUNT:POLARity?
Arguments	<p>POSitive indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.</p> <p>NEGAtive indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.</p> <p>EITHer triggers on a runt of either polarity.</p>
Examples	<p>TRIGger:A:RUNT:POLARity NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.</p> <p>TRIGger:A:RUNT:POLARity? might return :TRIGGER:A:RUNT:POLARITY POSITIVE indicating that the polarity of the A pulse runt trigger is positive.</p>

TRIGger:A:RUNT:SOURce

This command specifies the source waveform for the runt trigger.

NOTE. *Digital channels are not supported as runt trigger sources.*

Group	Trigger
Syntax	TRIGger:A:RUNT:SOURce {CH<x>} TRIGger:A:RUNT:SOURce?
Arguments	<p>CH<x> specifies the analog channel number to use as the source waveform for the runt trigger. To specify the threshold levels when using CH<x> as the source, use TRIGger:A:LOWerthreshold:CH<x> and TRIGger:A:UPPerthreshold:CH<x>.</p>
Examples	<p>TRIGger:A:RUNT:SOURce CH4 sets channel 4 as the trigger source.</p> <p>TRIGger:A:RUNT:SOURce? might return :TRIGGER:A:RUNT:SOURCE CH2 indicating that channel 2 is the trigger source.</p>

TRIGger:A:RUNT:WHEn

This command specifies the type of pulse width the trigger checks for when it detects a runt.

Group Trigger

Syntax TRIGger:A:RUNT:WHEn {LESSthan|MOREthan|EQua1|UNEQua1|OCCURS}
TRIGger:A:RUNT:WHEn?

Related Commands [TRIGger:A:RUNT:WIDth](#)

Arguments OCCURS sets the instrument to trigger if a runt signal of any detectable width occurs.

LESSthan sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the [TRIGger:A:RUNT:WIDth](#) command.

MOREthan sets the instrument to trigger if the a runt pulse is detected with width greater than the time set by the [TRIGger:A:RUNT:WIDth](#) command.

EQua1 sets the instrument to trigger if a runt pulse is detected with width equal to the time period specified in [TRIGger:A:RUNT:WIDth](#) within a $\pm 5\%$ tolerance.

NOTEQua1 sets the instrument to trigger if a runt pulse is detected with width greater than or less than (but not equal to) the time period specified in [TRIGger:A:RUNT:WIDth](#) within a $\pm 5\%$ tolerance.

Examples TRIGger:A:RUNT:WHEn MORETHAN sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

TRIGger:A:RUNT:WHEn? might return :TRIGGER:A:PULSE:RUNT:WHEn OCCURS indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.

TRIGger:A:RUNT:WIDth

This command specifies the width, in seconds, for a runt trigger.

Group Trigger

Syntax TRIGger:A:RUNT:WIDth <NR3>
TRIGger:A:RUNT:WIDth?

Related Commands [TRIGger:A:RUNT:WHEn](#)

Arguments <NR3> is a floating point number that specifies the minimum width, in seconds.

Examples TRIGger:A:RUNT:WIDTh 15E-6 sets the minimum width of the pulse runt trigger to 15 μ s.

TRIGger:A:RUNT:WIDTh? might return :TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09 indicating that the minimum width of a pulse runt trigger is 2 ns.

TRIGger:A:SETHold:CLOCK:EDGE

This command specifies the clock edge polarity for setup and hold triggering.

Group Trigger

Syntax TRIGger:A:SETHold:CLOCK:EDGE {FALL|RISE}
TRIGger:A:SETHold:CLOCK:EDGE?

Arguments FALL specifies polarity as the clock falling edge.
RISE specifies polarity as the clock rising edge.

Examples TRIGger:A:SETHold:CLOCK:EDGE RISE specifies the polarity as the clock rising edge.

TRIGger:A:SETHold:CLOCK:EDGE? might return :TRIGGER:A:SETHOLD:CLOCK:EDGE RISE indicating that polarity is specified as the clock rising edge.

TRIGger:A:SETHold:CLOCK:SOURce

This command specifies the clock source for the setup and hold triggering. You cannot specify the same source for both clock and data.

Group Trigger

Syntax TRIGger:A:SETHold:CLOCK:SOURce {CH<x>|DCH<x>_D<x>}
TRIGger:A:SETHold:CLOCK:SOURce?

Arguments	<p>CH<x> specifies the analog channel to use as the clock source waveform.</p> <p>DCH<x>_D<x> specifies the digital channel to use as the clock source waveform. The supported digital channel value is 1. The supported digital bit values are 0 to 15.</p>
Examples	<p>TRIGger:A:SETHold:CLOCK:SOURce CH1 specifies channel 1 as the clock source for a setup and hold trigger operation.</p> <p>TRIGger:A:SETHold:CLOCK:SOURce? might return :TRIGGER:A:SETHOLD:CLOCK:SOURCE CH4 indicating that channel 4 is the clock source for a setup and hold trigger operation.</p>

TRIGger:A:SETHold:HOLDTime

This command specifies the hold time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Hold Time.

Group	Trigger
Syntax	<p>TRIGger:A:SETHold:HOLDTime <NR3> TRIGger:A:SETHold:HOLDTime?</p>
Arguments	<p><NR3> is a floating point number that specifies the hold time setting, in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.</p>
Examples	<p>TRIGger:A:SETHold:HOLDTime 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.</p> <p>TRIGger:A:SETHold:HOLDTime? might return :TRIGGER:A:SETHOLD:HOLDTIME 2.0000E-09 indicating that the current hold time for the setup and hold trigger is 2 ns.</p>

TRIGger:A:SETHold:SETTime

This command specifies the setup time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Setup Time.

Group	Trigger
--------------	---------

Syntax	TRIGger:A:SETHold:SETTime <NR3> TRIGger:A:SETHold:SETTime?
Arguments	<NR3> is a floating point number that specifies the setup time for setup and hold violation triggering.
Examples	TRIGger:A:SETHold:SETTime 3.0E-6 specifies that the current setup time for setup and hold trigger is 3 μ s. TRIGger:A:SETHold:SETTime? might return :TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09 indicating that the current setup time for setup and hold trigger is 2 ns.

TRIGGER:A:SETHOLDLOGICVAL:DCH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger.

Group	Search and Mark
Syntax	TRIGGER:A:SETHOLDLOGICVAL:DCH<x>_D<x> {INCLude DONTInCLude} TRIGGER:A:SETHOLDLOGICVAL:DCH<x>_D<x>?
Arguments	DCH<x>_D<x> specifies the digital channel to use as the clock source waveform. The supported digital channel value is 1. The supported digital bit values are 0 to 15. INCLude specifies including the specified digital channel SETHOLD inputs. DONTInCLude specifies not including the specified digital channel SETHOLD inputs.
Examples	TRIGGER:A:SETHOLDLOGICVAL:DCH1_D0 INCLude specifies including the specified digital channel SETHOLD inputs. TRIGGER:A:SETHOLDLOGICVAL:DCH1_D0? might return TRIGGER:A:SETHOLDLOGICVAL:DCH1_D0 DONTINCLude indicating the specified digital channel SETHOLD inputs will not be included.

TRIGger:A:TIMEOut:POLarity

When triggering using the TIMEOut trigger type, this commands specifies the polarity to be used.

Group Trigger

Syntax TRIGger:A:TIMEOut:POLarity {STAYSHigh|STAYSLow|EITHer}
TRIGger:A:TIMEOut:POLarity?

Related Commands [TRIGger:A:TIMEOut:SOURce](#)
[TRIGger:A:TIMEOut:TIME](#)

Arguments STAYSHigh — Trigger when the signal stays high during the timeout time specified by the command [TRIGger:A:TIMEOut:TIME](#).
STAYSLow — Trigger when the signal stays low during the timeout time specified by the command [TRIGger:A:TIMEOut:TIME](#).
EITHer — Trigger when the signal is either high or low during the timeout time specified by the command [TRIGger:A:TIMEOut:TIME](#).

Examples TRIGger:A:TIMEOut:POLarity STAYSHigh specifies to trigger when the signal stays high during the timeout time when triggering using the TIMEOut trigger type.
TRIGger:A:TIMEOut:POLarity? might return EITHER, indicating that the signal stays either high or low during the timeout time.

TRIGger:A:TIMEOut:SOURce

When triggering using the TIMEOut trigger type, this command specifies the source. The available sources are live channels and digital channels. The default is channel 1. The timeout trigger type is selected using [TRIGger:A:TYPE TIMEOut](#).

Group Trigger

Syntax TRIGger:A:TIMEOut:SOURce {CH<x>|DCH<x>_D<x>}
TRIGger:A:TIMEOut:SOURce?

Related Commands [SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOURce](#)

[TRIGger:A:TIMEOut:TIME](#)

[TRIGger:A:TIMEOut:POLarity](#)

- Arguments** CH<x> specifies an analog channel as the timeout trigger source.
- DCH<x>_D<x> specifies a digital channel as the timeout trigger source. The supported digital channel value is 1. The supported digital bit values are 0 to 15.
- Examples** TRIGger:A:TIMEOut:SOUrce CH4 specifies to use channel 4 as the source for the timeout trigger.
- TRIGger:A:TIMEOut:SOUrce? might return CH1, indicating that channel 1 has been set as the source for the timeout trigger.

TRIGger:A:TIMEOut:TIME

When triggering using the TIMEOUT trigger type, this command specifies the timeout time, in seconds. This command is equivalent to selecting Timeout from the Trig menu and setting a value for Time Limit. The timeout trigger type is selected using [TRIGger:A:TYPE](#) TIMEOUT

Group Trigger

Syntax TRIGger:A:TIMEOut:TIME <NR3>
TRIGger:A:TIMEOut:TIME?

Related Commands [TRIGger:A:TIMEOut:POLarity](#)
[TRIGger:A:TIMEOut:SOUrce](#)

Arguments <NR3> is a floating point number that specifies the timeout time, in seconds.

Examples :TRIGger:A:TIMEOut:TIME 4.0E-9 specifies the timeout time of 4.0 nsec.
:TRIGger:A:TIMEOut:TIME? might return 8.0000E-9.

TRIGger:A:TYPE

This command sets or queries the type of A trigger.

Group Trigger

Syntax TRIGger:A:TYPE {EDGE| WIDTH| RISE| FALL| TIMEOut| RUNt| LOGIC| SETHoId| BUS}
TRIGger:A:TYPE?

Arguments EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the TRIGger:A:EDGE commands.

WIDTH specifies that the trigger occurs when a pulse with a specified width is found.

RISE specifies that the trigger occurs when a pulse with a specified rise is found.

FALL specifies that the trigger occurs when a pulse with a specified fall is found.

TIMEOut specifies that a trigger occurs when a pulse with the specified timeout is found.

RUNt specifies that a trigger occurs when a pulse with the specified parameters is found.

LOGIC specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:A:LOGIC commands.

SETHoId specifies that a trigger occurs when a signal is found that meets the setup and hold parameters.

BUS specifies that a trigger occurs when a signal is found that meets the specified bus setup parameters.

Examples TRIGGER:A:TYPE EDGE sets the A trigger type to EDGE.

TRIGGER:A:TYPE? might return :TRIGGER:A:TYPE RUNT indicating that the A trigger type is a runt trigger.

TRIGger:A:UPPerthreshold:CH<x>

This command sets or queries the specified channel upper trigger level. The CH<x> range is 1 to 4 and depends on the number of analog channels on your instrument.

Group Trigger

Syntax TRIGger:A:UPPerthreshold:CH<x> <NR3>
TRIGger:A:UPPerthreshold:CH<x>?

Related Commands [TRIGger:A:LOWerthreshold:CH<x>](#)

Arguments	<NR3> specifies the trigger level in user units (usually volts).
Examples	<p>TRIGGER:A:UPPERTHRESHOLD:CH1 1.3 This command sets the A trigger level for Channel 1 to 1.3 V level.</p> <p>TRIGGER:A:UPPERTHRESHOLD:CH2? might return TRIGGER:A:UPPERTHRESHOLD:CH2 1.3000E+00, indicating that the A trigger level for Channel 2 is set to 1.3 V.</p>

TRIGger:A:HOLDoff:TIME

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Mode & Holdoff from the Trig menu, selecting Time, and then setting the desired Holdoff Time.

Group	Trigger
Syntax	<p>TRIGger:A:HOLDoff:TIME <NR3></p> <p>TRIGger:A:HOLDoff:TIME?</p>
Arguments	<NR3> specifies the holdoff time in seconds. The range is from 0 seconds through 10 seconds.
Examples	<p>TRIGGER:A:HOLDOFF:TIME 10 sets the A trigger holdoff time to 10 s.</p> <p>TRIGGER:A:HOLDOFF:TIME? might return :TRIGGER:A:HOLDOFFTIME 1.2000E-06, indicating that the A trigger holdoff time is set to 1.2 μs.</p>

TRIGger:A:MODE

This command sets or queries the A trigger mode. This command is equivalent to pushing the Mode button on the front panel.

Group	Trigger
Syntax	<p>TRIGger:A:MODE {AUTO NORMa1}</p> <p>TRIGger:A:MODE?</p>
Arguments	<p>AUTO generates a trigger if one is not detected within a specified time period.</p> <p>NORMa1 waits for a valid trigger event.</p>

Examples TRIGGER:A:MODE NORMAL sets the trigger mode to normal.
TRIGGER:A:MODE? might return TRIGGER:A:MODE AUTO, indicating that the trigger mode is auto.

TRIGger:AUXLevel

This command sets or queries the Auxiliary Input voltage level to use for an edge trigger.

Group Trigger

Syntax TRIGger:AUXLevel {<NR3>|ECL|TTL}
TRIGger:AUXLevel?

Arguments <NR3> is trigger level in Volts.
ECL sets trigger level to -1.3 Volts.
TTL sets trigger level to 1.4 Volts.

Returns <NR3> is trigger level in Volts.

Examples TRIGger:AUXLevel 1.2 sets trigger level for the Auxiliary Input to 1.2 Volts.

TRIGger:HYSteresis:USER:STATe

This command enables or disables user-defined trigger hysteresis.

Group Trigger

Syntax TRIGger:HYSteresis:USER:STATe {ON|OFF|1|0}
TRIGger:HYSteresis:USER:STATe?

Related Commands [TRIGger:HYSteresis:USER:VALue](#)

Arguments ON enables user-defined trigger hysteresis.
OFF disables user-defined trigger hysteresis.
1 enables user-defined trigger hysteresis.

0 disables user-defined trigger hysteresis.

Examples TRIGGER:HYSTERESIS:USER:STATE ON enables user-defined trigger hysteresis.
 TRIGGER:HYSTERESIS:USER:STATE? might return
 TRIGGER:HYSTERESIS:USER:STATE 0, indicating that the user-defined trigger hysteresis is disabled.

TRIGGER:HYSTERESIS:USER:VALUE

This command sets or returns the height of the user-defined trigger hysteresis zone in volts.

Group Trigger

Syntax TRIGGER:HYSTERESIS:USER:VALUE <NR1>
 TRIGGER:HYSTERESIS:USER:VALUE?

Related Commands [TRIGGER:HYSTERESIS:USER:STATE](#)

Arguments <NR1> sets the height of the hysteresis zone in volts.

Examples TRIGGER:HYSTERESIS:USER:VALUE 27.0e-3 sets the height of the hysteresis zone to 27 mV.
 TRIGGER:HYSTERESIS:USER:VALUE? might return
 TRIGGER:HYSTERESIS:USER:VALUE 27.0e-3, indicating that the height of the hysteresis zone is 27 mV.

TRIGGER:STATE? (Query Only)

This query-only command returns the current state of the triggering system.

Group Trigger

Syntax TRIGGER:STATE?

Related Commands [TRIGGER:A:MODE](#)

Returns ARMED indicates that the instrument is acquiring pretrigger information.

AUTO indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.

READY indicates that all pretrigger information is acquired and that the instrument is ready to accept a trigger.

SAVE indicates that the instrument is in save mode and is not acquiring data.

TRIGGER indicates that the instrument triggered and is acquiring the post trigger information.

Examples TRIGGER:STATE? might return TRIGGER:STATE ARMED, indicating that the pretrigger data is being acquired.

*TST? (Query Only)

Tests (self-test) the interface and returns a 0.

Group Miscellaneous

Syntax *TST?

Examples *TST? always returns 0.

UNDO (No Query Form)

Reverts the instrument settings to a state before the previous command or user interface action.

Group Miscellaneous

Syntax UNDO

Examples UNDO reverts the instrument settings to a state before the previous command or user interface action.

UNLock (No Query Form)

This command (no query form) unlocks the front panel controls only. To unlock the front panel controls and the touch screen use the `LOCK NONE` command. The command `TOUCHSCREEN:STATE ON` enables the touch screen only.

NOTE. *If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.*

Group	Miscellaneous
Syntax	UNLock ALL
Related Commands	LOCK
Arguments	ALL specifies that all front panel buttons and knobs are unlocked.
Examples	UNLOCK ALL unlocks all front panel buttons and knobs.

USBDevice:CONFigure

This command may be used to configure the rear USB port to be off or enabled as a USBTMC device. Users should be cautious using this command via the USBTMC interface as a change to the configuration of this interface from a USBTMC device will cause USBTMC communication to cease. It is intended to be used via the Ethernet interface to control the USB device interface.

Group	Miscellaneous
Syntax	USBDevice:CONFigure {DISabled USBTmc} USBDevice:CONFigure?
Arguments	DISabled will disable the rear USB port. USBTmc enables the rear USB port.

Examples USBDEVICE:CONFIGURE DISABLED will disable the rear USB port.

USBDEVICE:CONFIGURE? might return :USBDEVICE:CONFIGURE USBTMC indicating the USB port is enabled.

VERBose

This command sets or queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

NOTE. *This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).*

Group Miscellaneous

Syntax VERBose {<NR1>|OFF|ON}

Related Commands **HEADer**
 *LRN?
 SET?

Arguments <NR1> = 0 disables Verbose, any other value enables Verbose.

OFF sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.

ON sets the Verbose state to true, which returns full-length keywords for applicable setting queries.

A 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

Examples VERBOSE ON sets the Verbose state to true and return the full length keyword for the applicable setting queries.

VERBOSE? might return :VERBOSE OFF, indicating that the Verbose state is set to false and return the minimum-length keywords for the applicable setting queries.

With :HEADer ON and :VERBose ON, the :ACQuire:MODE? query might return: :ACQUIRE:MODE SAMPLE

With :HEADer ON and :VERBose OFF, the :ACQuire:MODE? query might return: :ACQ:MOD SAM

With `:HEADer OFF` and `:VERBoSe ON`, the `:ACQuiRe:MODE?` query might return: `SAMPLE`

With `:HEADer OFF` and `:VERBoSe OFF`, the `:ACQuiRe:MODE?` query might return: `SAM`

VERTical:DESKew:FROM:CUSTOMPROPAGation

This command sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

Group	Vertical
Syntax	<code>VERTical:DESKew:FROM:CUSTOMPROPAGation <NR3></code> <code>VERTical:DESKew:FROM:CUSTOMPROPAGation?</code>
Arguments	<code><NR3></code> is a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.
Examples	<code>VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.25e-9</code> sets the custom delay to 250 ps. <code>VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION?</code> might return <code>:VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.0E+0</code> indicating the custom delay is 0.0 s.

VERTical:DESKEW:FROM:SOURce

This command sets or queries the source channel for performing channel-to-channel deskew adjustment. Sources can be any of the analog channels.

Group	Vertical
Syntax	<code>VERTical:DESKEW:FROM:SOURce CH<x></code> <code>VERTical:DESKEW:FROM:SOURce?</code>
Arguments	The vertical deskew source.

Examples `VERTICAL:DESKEW:FROM:SOURCE CH1` sets channel 1 as the from source.
`VERTICAL:DESKEW:FROM:SOURCE?` might return
`:VERTICAL:DESKEW:FROM:SOURCE CH2` indicating channel 2 is the from source.

VERTical:DESKEW:STATIC (No Query Form)

This command executes static deskew using the deskew settings.

Group Vertical

Syntax `VERTical:DESKEW:STATIC EXECute`

Arguments `EXECute` will execute static deskew using the deskew settings.

Examples `VERTICAL:DESKEW:STATIC EXECUTE` will execute static deskew using the deskew settings.

VERTical:DESKew:TO:CUSTOMPROPagation

This command sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Group Vertical

Syntax `VERTical:DESKew:TO:CUSTOMPROPagation <NR3>`

Arguments `<NR3>` is a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Examples `VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.25e-9` sets the custom propagation to 250 ps.
`VERTICAL:DESKEW:TO:CUSTOMPROPAGATION?` might return
`:VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.0E+0` indicating the custom propagation is set to 0.0 ns.

VERTical:DESKEW:TO:SOURce

This command sets or queries the target channel for performing channel-to-channel deskew adjustment. Target sources can be any of the live analog channels.

Group	Vertical
Syntax	VERTical:DESKEW:TO:SOURce CH<x> VERTical:DESKEW:TO:SOURce?
Arguments	Arguments are the live analog channels.
Examples	VERTICAL:DESKEW:TO:SOURCE CH4 sets the deskew to source is channel 4. VERTICAL:DESKEW:TO:SOURCE? might return :VERTICAL:DESKEW:TO:SOURCE CH2 indicating the deskew to source is channel 2.

*WAI (No Query Form)

The *WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

Group	Status and Error
Syntax	*WAI
Related Commands	BUSY? *OPC
Examples	*WAI prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

WAVFrm? (Query Only)

This query-only command provides the Tektronix standard waveform query which returns the waveform preamble followed by the waveform data for the source specified by :DATA:SOURce using the :DATA settings for encoding, width, and so forth.

Group Waveform Transfer

Syntax WAVFrm?

Related Commands [CURVe?](#)
[DATA:SOURce](#)
[WFMOupre?](#)

Examples WAVFRM? might return the waveform data as: WFMOUTPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1; BYT_OR MSB;ENCDG ASC;NR_PT 500;PT_FMT Y; PT_ORDER LINEAR;PT_OFF 0;XINCR 400.0000E-12; XZERO 0.0000;XUNIT "s";YMULT 4.0000E-3; YOFF 0.0000;YZERO 0.0000;YUNIT "V"; WFID "Ch1,DC coupling, 100.0mV/div, 200.0ns/div, 5000 points,Samp1 mode".

WFMOupre? (Query Only)

This query-only command queries the waveform formatting data for the waveform specified by the [DATA:SOURce](#) command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCDg, BN_Fmt, BYT_Or, BYT_Nr, BIT_Nr. The interpretation components are derived from the DATA:SOURce specified waveform.

Group Waveform Transfer

Syntax WFMOupre?

Examples WFMOUTPRE? might return the waveform formatting data as: WFMOUTPRE:BYT_NR 2;BIT_NR 16;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 200.0mV/div, 10.00us/div, 1250 points, Sample mode";NR_PT 1000;PT_FMT Y;XUNIT "s";XINCR 80.0000E-9;XZERO


```
0.0000;PT_OFF 625;YUNIT "V";YMULT 31.2500E-6;YOFF
0.0000;YZERO 0.0000;NR_FR 3.
```

WFMOutpre:ASC_Fmt? (Query Only)

This query returns the format for ASCII data transferred from the instrument. No command form is provided as the format is determined by the data source type. Some waveforms are normalized vector data where the data points are 8-byte doubles in floating point format whereas other formats are 1-byte or 2-byte integers.

Group Waveform Transfer

Syntax WFMOutpre:ASC_Fmt?

Related Commands [DATA:SOURce](#)
[WFMOutpre:BN_Fmt](#)
[WFMOutpre:ENCdg](#)

Returns FP represents floating point ASCII data. The waveforms are normalized vector data where the data points are 8-byte doubles in floating point format.
 INTEGER represents signed integer ASCII data. The waveform data are 1-byte or 2-byte integers.

Examples WFMOutpre:ASC_Fmt? might return WFMOUTPRE:ASC_FMT INTEGER indicating the ASCII format is integer.

WFMOutpre:BIT_Nr

This command sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the [DATA:SOURce](#) command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when [WFMOutpre:ENCdg](#) is set to BIN and [WFMOutpre:BN_Fmt](#) is set to either RI or RP.

Group Waveform Transfer

Syntax	WFMOutpre:BIT_Nr <NR1> WFMOutpre:BIT_Nr?
Related Commands	DATA:SOURce WFMOutpre:BN_Fmt WFMOutpre:ENCdg
Arguments	<NR1> number of bits per data point can be 8 or 16.
Examples	WFMOUTPRE:BIT_NR 16 sets the number of bits per waveform point to 16 for incoming RI and RP binary format data. WFMOUTPRE:BIT_NR? might return WFMOUTPRE:BIT_NR 8, indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

WFMOutpre:BN_Fmt

This command sets or queries the format of binary data for outgoing waveforms specified by the [DATA:SOURce](#) command.

Group	Waveform Transfer
Syntax	WFMOutpre:BN_Fmt {RI RP FP} WFMOutpre:BN_Fmt?
Related Commands	DATA:SOURce
Arguments	RI specifies signed integer data point representation. RP specifies positive integer data point representation. FP specifies floating point representation.
Examples	WFMOUTPRE:BN_FMT FP specifies that outgoing waveform data will be in single-precision binary floating point format. WFMOUTPRE:BN_FMT? might return WFMOUTPRE:BN_FMT RI, indicating that the outgoing waveform data is currently in signed integer format.

WFMOutpre:BYT_Nr

This command sets or queries the binary field data width (bytes per point) for the waveform specified by the [DATA:SOURce](#) command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when [WFMOutpre:ENCdg](#) is set to BIN, and [WFMOutpre:BN_Fmt](#) is set to either RI or RP.

Group Waveform Transfer

Syntax WFMOutpre:BYT_Nr <NR1>
WFMOutpre:BYT_Nr?

Related Commands [DATA:SOURce](#)
[WFMOutpre:BN_Fmt](#)
[WFMOutpre:ENCdg](#)

Arguments <NR1> is the number of bytes per data point and can be 1, 2 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 8 bytes per waveform point indicate pixel map (fast acquisition) data.

Examples WFMOUTPRE:BYT_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.
WFMOUTPRE:BYT_NR? might return WFMOUTPRE:BYT_NR 2, indicating that there are 2 bytes per outgoing waveform data point.

WFMOutpre:BYT_Or

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when [WFMOutpre:ENCdg](#) is set to BIN.

Group Waveform Transfer

Syntax WFMOutpre:BYT_Or {LSB|MSB}
WFMOutpre:BYT_Or?

Related Commands [WFMOutpre:ENCdg](#)

Arguments LSB specifies that the least significant byte will be transmitted first.
 MSB specifies that the most significant byte will be transmitted first.

Examples WFMOUTPRE:BYT_OR MSB sets the most significant outgoing byte of waveform data to be transmitted first.

 WFMOUTPRE:BYT_OR? might return WFMOUTPRE:BYT_OR LSB, indicating that the least significant data byte will be transmitted first.

WFMOutpre:CENTERFREQUENCY? (Query Only)

This query returns the center frequency of the outgoing waveform. The center frequency is a precise frequency at the center of the display of the frequency domain graticule. In many applications, it is a carrier frequency.

Group Waveform Transfer

Syntax WFMOutpre:CENTERFREQUENCY?

Related Commands [DATA:SOURce](#)

Returns Returns the center frequency of the outgoing waveform, in Hertz.

Examples WFMOutpre:CENTERFREQUENCY? might return
 WFMOutpre:CENTERFREQUENCY 1.0E+9, indicating a center frequency of 1 GHz.

WFMOutpre:DOMAIN? (Query Only)

This query returns the domain of the outgoing waveform, either TIME or FREQUENCY. If the domain is TIME, it indicates that the data is to be treated as integer information. If the domain is FREQUENCY, it indicates that the data is to be treated as floating point information.

Group Waveform Transfer

Syntax WFMOutpre:DOMAIN?

Related Commands	DATA:SOURce WFMOutpre:BN_Fmt WFMOutpre:ENCdg
Returns	Returns the domain of the outgoing waveform.
Examples	<p>WFMOUTPRE:DOMAIN? might return WFMOUTPRE:DOMAIN TIME, indicating that the outgoing waveform is a time domain trace.</p> <p>WFMOutpre:DOMain? might return WFMOutpre:DOMain FREQuency, indicating that the outgoing waveform is one of the four Spectrum View frequency domain traces (CH<x>_SV_NORMAl, CH<x>_SV_AVErage, CH<x>_SV_MINHOLD, or CH<x>_SV_MAXHOLD).</p>

WFMOutpre:ENCdg

This command sets or queries the type of encoding for outgoing waveforms.

Group	Waveform Transfer
Syntax	WFMOutpre:ENCdg {ASCIi BINary} WFMOutpre:ENCdg?
Related Commands	DATA:ENCdg WFMOutpre:BYT_Nr WFMOutpre:BYT_Or WFMOutpre:BIT_Nr WFMOutpre:BN_Fmt
Arguments	<p>ASCIi specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.</p> <p>BINary specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT_Nr, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt and WFMOutpre:BYT_Or.</p>
Examples	WFMOUTPRE:ENCDG ASCIi specifies that the outgoing waveform data will be sent in ASCII format.

WFMOU

```
PRE:ENCDG?
```

 might return `WFMOU

```
PRE:ENCDG
```

 BINary`, indicating that outgoing waveform data will be sent in binary format.

WFMOU ``` pre:NR_Pt? ``` (Query Only)

This query-only command returns the number of points for the `DATA:SOURCE` waveform that will be transmitted in response to a `CURVe??` query.

Group Waveform Transfer

Syntax `WFMOU

```
pre:NR_Pt?
````

Related Commands [CURVe?](#)
[DATA](#)
[DATA:START](#)
[DATA:STOP](#)
[SAVE:WAVEform](#)
[SAVEON:WAVEform:FILEFormat](#)

Examples `WFMOU

```
PRE:NR_PT?
```` might return `WFMOU

```
PRE:NR_PT
```

 5000`, indicating that there are 5000 data points to be sent.

WFMOU ``` pre:PT_Fmt? ``` (Query Only)

This query-only command returns the point format for the waveform specified by the `DATA:SOURCE` command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the `CURVe?` data points.

An error is reported if the `DATA:SOURCE` waveform does not exist.

Group Waveform Transfer

Syntax `WFMOU

```
pre:PT_Fmt?
````

Related Commands [CURVe?](#)
[DATA:SOURCE](#)

Examples `WFMOUTPRE:PT_FMT?` might return `WFMOutpre:PT_Fmt ENV`, indicating that the waveform data is a series of min-max pairs.

WFMOutpre:PT_Off? (Query Only)

This query-only command returns the trigger point relative to [DATA:START](#) for the waveform specified by the [DATA:SOURce](#) command.

NOTE. *This returned value is the point immediately following the actual trigger.*

Group Waveform Transfer

Syntax `WFMOutpre:PT_Off?`

Related Commands [DATA:SOURce](#)
[DATA:START](#)
[WFMOutpre:XZEro?](#)

Examples `WFMOUTPRE:PT_OFF?` might return `WFMOUTPRE:PT_OFF 251` specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:PT_ORder? (Query Only)

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 500 (vertical) by 1000 (horizontal) point bitmap. Each point represents display intensity for that screen location. Only `CURVe?` query functions are allowed on Fast Acquisition waveforms. When the `WFMOutpre:PT_ORder?` query returns `Row`, this indicates that the source is a Fast Acquisition waveform (and that each of 1000 possible horizontal columns being transmitted contains 500 vertical points. Note: 500 might vary based on how many channels enabled from same FastAcq group.) and waveform points are transmitted in the following order: left to right; then top to bottom.

When the `WFMOutpre:PT_ORder?` query returns `Linear`, this indicates that the source is not a Fast Acquisition waveform (and that each horizontal column being sent contains only one vertical point). Note that waveform points are transmitted in the following order: top to bottom, then left to right.

Group Waveform Transfer

Syntax `WFMOutpre:PT_Order?`

Related Commands [DATA:SOURce](#)

Examples `WFMOUTPRE:PT_ORDER?` might return `WFMOUTPRE:PT_Order ROW`, specifying that the waveform designated by the `DATA:SOURce` waveform is a Fast Acquisition waveform and `CURve?` is returning the PIXmap data for that source.

`WFMOUTPRE:PT_ORDER?` might return `WFMOUTPRE:PT_Order LINEAR`, specifying that the source waveform is a non-Fast Acquisition waveform and `CURve?` is returning the Vector data for that source.

WFMOutpre:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting `DATA:RESample`. Setting the `WFMOutpre:RESample` value causes the corresponding `DATA` value to be updated and vice versa.

Group Waveform Transfer

Syntax `WFMOutpre:RESample <NR1>`
`WFMOutpre:RESample?`

Related Commands [DATA:RESample](#)

Arguments `<NR1>` is the resampling rate. The default value is 1, which means every sample is returned. A value of 2 returns every other sample, while a value of 3 returns every third sample, and so on.

Examples `WFMOUTPRE:RESAMPLE 1` sets the resampling to return all waveform data samples.

`WFMOUTPRE:RESAMPLE?` might return `WFMOUTPRE:RESAMPLE 4`, indicating that every fourth sample of the waveform data is returned.

WFMOutpre:SPAN? (Query Only)

This query returns the frequency span of the outgoing waveform. The span is the range of frequencies you can observe around the center frequency.

Group Waveform Transfer

Syntax WFMOutpre:SPAN?

Related Commands [DATA:SOURce](#)

Returns Returns the frequency span of the outgoing waveform, in Hertz.

Examples RF:SPAN? might return RF:SPAN 10.0000E+6, indicating the span is 10 MHz.

WFMOutpre:WFId? (Query Only)

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the [DATA:SOURce](#) command.

An error is reported if the DATA:SOURce waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:WFId?

Related Commands [DATA:SOURce](#)

Returns <QString> contains the following comma-separated fields documented in the following tables:

Table 2-48: Waveform Suffixes

| Field | Description | Examples |
|----------|---|--|
| Source | The source identification string as it appears in the front panel scale factor readouts. | "CH1-8"
"Math<x>"
"Ref<x>" |
| Coupling | A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math). | "AC coupling"
"DC coupling"
"GND coupling" |

Table 2-48: Waveform Suffixes (cont.)

| Field | Description | Examples |
|------------------|--|--|
| Vert Scale | A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units. | "100.0 mV/div"
"20.00 dB/div"
"45.00 deg/div"
"785.4 mrad/div"
"500.0 μ Vs/div"
"10.00 kV/s/div"
"200.0 mV/div"
"50.00 unk/div" |
| Horiz Scale | A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units. | "100 ms/div"
"10.00 kHz/div"
"50.00 c/div" |
| Record Length | A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer. | "500 points"
"500000 points" |
| Acquisition Mode | A string describing the mode used to acquire the waveform. | "Sample mode"
"Pk Detect mode"
"Hi Res mode"
"Envelope mode"
"Average mode" |
| Center Frequency | A string containing the center frequency value of the waveform. | "2.400GHz" |
| Span | A string containing the frequency span value of the waveform. | "20.00MHz" |
| Reference Level | A string containing the reference level value of the waveform. | "1.000mW" |

Examples WFMOUTPRE:WFID? might return WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0 μ s/div,500 points, Hi Res mode".

WFMOutpre:WFMTYPE? (Query Only)

This query returns the type of the outgoing waveform.

The type of waveform that is being transferred in turn determines in which window it will display on the instrument screen: (the Waveform View window or Spectrum View window). Both the Analog and RF_TD arguments specify the time domain window; the RF_RD argument specifies the frequency domain window.

Group Waveform Transfer

Syntax `WFMOutpre:WFMTYPE?`

Related Commands [DATA:SOURce](#)

Returns Returns the type of the outgoing waveform:
 ANALOG indicates a Channel or Math waveform.
 SV_FD indicates a Spectrum View RF frequency domain waveform.
 RF_TD indicates an RF time domain waveform.

Examples `WFMOutpre:WFMTYPE?` might return `WFMOutpre:WFMTYPE CH<x>_SV_FD`, which indicates that the outgoing waveform is an RF frequency domain waveform (`CH<x>_SV_AVErge`, `CH<x>_SV_MAXHold`, `CH<x>_SV_MINHold`, or `CH<x>_SV_NORMal`).

WFMOutpre:XINcr? (Query Only)

This query-only command returns the horizontal point spacing in units of `WFMOutpre:XUNit` for the waveform specified by the [DATA:SOURce](#) command. This value corresponds to the sampling interval.

An error is reported if the `DATA:SOURce` waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:XINcr?`

Related Commands [DATA:SOURce](#)
[WFMOutpre:XUNit?](#)

Examples `WFMOUPRE:XINCR?` might return `WFMOUPRE:XINCR 10.0000E-6`, indicating that the horizontal sampling interval is 10 μ s/point (500 μ s/div).

WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the [DATA:SOURce](#) command.

An error is reported if the `DATA:SOURce` waveform does not exist.

| | |
|-------------------------|---|
| Group | Waveform Transfer |
| Syntax | WFMOutpre:XUNit? |
| Related Commands | DATA:SOURce |
| Examples | WFMOUTPRE:XUNIT? might return WFMOUTPRE:XUNIT "HZ", indicating that the horizontal units for the waveform are in Hertz. |

WFMOutpre:XZEro? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the [DATA:SOURce](#) command. This value is in units of WFMOutpre:XUNit.

An error is reported if the [DATA:SOURce](#) waveform does not exist.

NOTE. During steady state operation, when all control changes have settled and triggers are arriving on a regular basis, this is the only part of the preamble that changes on each acquisition.

| | |
|-------------------------|---|
| Group | Waveform Transfer |
| Syntax | WFMOutpre:XZEro? |
| Related Commands | DATA:SOURce
WFMOutpre:XUNit? |
| Examples | WFMOUTPRE:XZERO? might return WFMOUTPRE:XZERO 5.6300E-9, indicating that the trigger actually occurred 5.63 ns before the trigger sample. |

WFMOutpre:YMUIt? (Query Only)

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the [DATA:SOURce](#) command. For those formats in which [WFMOutpre:BYT_Nr](#) is important (all non-floating point formats), WFMOutpre:YMUIt? must take the location of the binary point implied by [BYT_NR](#) into consideration.

An error is reported if the `DATA:SOURce` waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YMULT?`

Related Commands [DATA:SOURce](#)

Examples `WFMOUTPRE:YMULT?` might return `WFMOUTPRE:YMULT 4.0000E-3`, indicating that the vertical scale for the corresponding waveform is 100 mV/div.

WFMOutpre:YOFF? (Query Only)

This query-only command returns the vertical offset of the source specified by [DATA:SOURce](#). For this instrument family, the value returned is always 0.0 as the offset is combined with the `:YZERO` value.

An error is reported if the [DATA:SOURce](#) waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YOFF?`

Related Commands [DATA:SOURce](#)
[WFMOutpre:BYT_Nr](#)

Examples `WFMOUTPRE:YOFF?` might return `WFMOUTPRE:YOFF -50.0000E+0`, indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.

WFMOutpre:YUNit? (Query Only)

This query-only command returns the vertical units for the waveform specified by the [DATA:SOURce](#) command.

An error is reported if the `DATA:SOURce` waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:YUNit?

Related Commands [DATA:SOURce](#)

Examples WFMOUTPRE:YUNIT? might return WFMOUTPRE:YUNIT "dB", indicating that the vertical units for the waveform are measured in decibels.

WFMOutpre:YZero? (Query Only)

This query-only command returns the combined vertical position and offset for the source waveform specified by [DATA:SOURce](#). This represents a departure from previous instruments where the :YZero value represented the vertical position in vertical units and the :YOff value represented the vertical offset in digitizing levels. For this instrument family, the value of :YOff is always 0.0.

An error is reported if the [DATA:SOURce](#) waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:YZero?

Related Commands [DATA:SOURce](#)
[WFMOutpre:YUNit?](#)

Examples WFMOUTPRE:YZERO? might return WFMOUTPRE:YZERO -100.0000E-3, indicating that vertical offset is set to -100 mV.

Status and Events

The instrument provides a status and event reporting system for the Ethernet and USB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

Registers

Overview

The registers in the event handling system fall into two functional groups:

- Status Registers contain information about the status of the instrument. They include the Standard Event Status Register (SESR).
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

The Standard Event Status Register (SESR). The SESR records eight types of events that can occur within the instrument. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

NOTE. *TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.*

Figure 3-1: The Standard Event Status Register (SESR)

Table 3-1: SESR Bit Functions

| Bit | Function | |
|---------|----------|--|
| 7 (MSB) | PON | Power On. Shows that the instrument was powered on. On completion, the diagnostic self tests also set this bit. |
| 6 | URQ | User Request. Indicates that an application event has occurred. *See note. |
| 5 | CME | Command Error. Shows that an error occurred while the instrument was parsing a command or query. |
| 4 | EXE | Execution Error. Shows that an error executing a command or query. |
| 3 | DDE | Device Error. Shows that a device error occurred. |
| 2 | QYE | Query Error. Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost. |
| 1 | RQC | Request Control. This is not used. |
| 0 (LSB) | OPC | operation complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command. |

The Status Byte Register (SBR). Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

Figure 3-2: The Status Byte Register (SBR)

Table 3-2: SBR Bit Functions

| Bit | Function | |
|---------|----------|---|
| 7 (MSB) | ———— | Not used. |
| 6 | RQS | Request Service. Obtained from a serial poll. Shows that the instrument requests service from the controller. |
| 6 | MSS | Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR. |
| 5 | ESB | Event Status Bit. Shows that status is enabled and present in the SESR. |
| 4 | MAV | Message Available. Shows that output is available in the Output Queue. |
| 3 | ———— | Not used. |

Table 3-2: SBR Bit Functions (cont.)

| Bit | Function |
|-----|-----------|
| 2 | Not used. |
| 1-0 | Not used. |

Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER). This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

Figure 3-3: The Device Event Status Enable Register (DESER)

The Event Status Enable Register (ESER). This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.

Figure 3-4: The Event Status Enable Register (ESER)

The Service Request Enable Register (SRER). This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

Figure 3-5: The Service Request Enable Register (SRER)

***PSC Command** The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending *PSC 0 lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

NOTE. *To enable the PON (Power On) event to generate a Service Request, send *PSC 0, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.*

Queues

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

Output Queue The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.



CAUTION. *When a controller sends a query, an <EOM>, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.*

Event Queue The Event Queue stores detailed information on up to 33 events. If than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

Event Handling Sequence

The following figure shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.

Figure 3-6: Status and Event Handling Process

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESR is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

Synchronization Methods

Overview Although most commands are completed almost immediately after being received by the instrument, some commands start a process that requires time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may take an extended period of time before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent. Furthermore, sometimes the result of an operation depends upon the result of an earlier operation. A first operation must complete before the next one is processed.

In order to handle these situations, the instrument status and event reporting system allows you to synchronize the operation of the instrument with your application program, using the Operation Complete function. Note, however, that only some operations are able to take advantage of this function; a table is provided below of commands that support this.

The following commands are used to synchronize the instrument functions using Operation Complete. See examples of how to use these commands later on in this section:

***OPC** — sending the *OPC command will set bit 0 of the SESR (Standard Events Status Register). The bit will only be set high when all pending operations that generate an OPC message have finished execution. (The SESR is queried using *ESR?) The *OPC? query form returns 1 only when all operations have completed, or a device clear is received.

***WAI** — prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete.

BUSY? — returns the status of operations: 1 means there are pending operations, 0 means that all pending operations are complete.

NOTE. Some OPC operations may require an extended period of time to complete or may never complete. For example, a single sequence acquisition may never complete when no trigger event occurs. You should be aware of these conditions and tailor your program accordingly by:

— setting the timeout sufficiently for the anticipated maximum time for the operation and

— handle a timeout appropriately by querying the SESR (*ESR?) and event queue (EVMsg? or ALLEv?).

NOTE. The *OPC command form can also be used to cause an SRQ to be generated upon completion of all pending operations. This requires that the ESB (Event Status Bit, bit 5) of the SRER (Service Request Enable Register) is set, and the OPC bit (bit 0) of the DESR (Device Event Status Enable Register) and the ESER (Event Status Enable Register) are set. (The SRER is set/queried using *SRE. The DESR is set/queried using DESE. The ESER is set/queried using *ESE.)

Only a subset of instrument operations support the Operation Complete function (OPC):

Table 3-3: instrument operations that can generate OPC

| Command | Conditions |
|--------------------------------------|---|
| ACQuire:STATE <non-zero NR1> ON RUN | Only when in single sequence acquisition mode |
| :AUTOset < EXECute > | |
| CALibrate:INTERNaI | |
| CALibrate:INTERNaI:STARt | |
| CALibrate:FActory STARt | |

Table 3-3: instrument operations that can generate OPC (cont.)

| Command | Conditions |
|---|--|
| CALibrate:FACTory CONTInue | |
| CALibrate:FACTory PREVious | |
| DIAG:STATE EXECute | |
| FACTory | |
| MEASUrement:MEAS<x> :RESUlts | When used in single sequence acquisition mode or during waveform recall. |
| RECALL:SETUp (<file as quoted string> FACTory) | |
| RECALL:WAVEform <.ISF or .CSV file >,< REF<x> > | |
| :RF:REFLevel AUTO | |
| *RST | |
| SAVe:IMAGe <file as quoted string> | |
| SAVe:SETUp <file as quoted string> | |
| SAVe:WAVEform < source wfm >, (< REF<x> > < file >) | |
| TEKSecure | |
| TRIGger:A SETLevel | |

Example of Acquiring and Measuring a Single-Sequence Waveform

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```

/** Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data */
ACQUIRE:STATE ON
/** Set up the measurement parameters */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/** Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?

```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.

Figure 3-7: Command Processing Without Using Synchronization

To ensure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program using *WAI, BUSY, *OPC, and *OPC?.

Figure 3-8: Processing Sequence With Synchronization**Example of Using the *OPC Command**

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.

Serial Poll Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* wait until the acquisition is complete before taking the
measurement. */
*OPC
while serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULTS:CURRENTacq:MEAN?
```

This technique requires less bus traffic than did looping on BUSY.

Service Request Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, the instrument will generate a Service Request.

The same command sequence using the *OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32

/* Set up the measurement parameters on the channel we're
about to sequence */ :MEASUREMENT:MEAS1:TYPE AMPLITUDE
:MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* wait until the acquisition is complete before taking the
measurement*/
*OPC
```

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
```

Example of Using the *OPC? Query

The *OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. The *OPC? query does not return until all pending OPC operations have completed. Therefore, your time-out must be set to a time at least as long as the longest expected time for the operations to complete.

The same command sequence using the *OPC? query for synchronization looks like this:

```
/* Set up single sequence acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
```

```
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* wait until the acquisition is complete before taking the
measurement*/
*OPC?
```

Wait for read from Output Queue.

```
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULTS:CURRENTacq:MEAN?
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

Example of Using the *WAI Command

The *WAI command forces completion of previous commands that generate an OPC message. No commands after the *WAI are processed before the OPC message(s) are generated

The same command sequence using the *WAI command for synchronization looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* wait until the acquisition is complete before taking
the measurement*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESULTS:CURRENTacq:MEAN?
```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all in-process OPC operations are complete. If the input buffer becomes full, the

controller will be unable to write commands to the buffer. This can cause a time-out.

Example of Using the BUSY Query

The BUSY? query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking
the measurement */
while BUSY? keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the *WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in bus traffic.

Reference waveforms

Measurements on references also support OPC when used in conjunction with a :RECALL:WAVEFORM command.

```
/* Load the waveform file */
:RECALL:WAVEFORM "C:/Digital8.wfm",REF1
*OPC?
/* Set up the measurement parameters on the reference */
:MEASUREMENT:MEAS1:TYPE PERIOD
:MEASUREMENT:MEAS1:SOURCE REF1
*OPC?
/* wait for read from Output Queue. */
/* Take amplitude measurement */
:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MEAN?
```

Messages

The information contained in the topics above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

No Event The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

Table 3-4: No Event Messages

| Code | Message |
|------|---|
| 0 | No events to report; queue empty |
| 1 | No events to report; new events pending *ESR? |

Command Error The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

Table 3-5: Command Error Messages (CME Bit 5)

| Code | Message |
|------|------------------------------|
| 100 | Command error |
| 101 | Invalid character |
| 102 | Syntax error |
| 103 | Invalid separator |
| 104 | Data type error |
| 105 | GET not allowed |
| 108 | Parameter not allowed |
| 109 | Missing parameter |
| 110 | Command header error |
| 112 | Program mnemonic too long |
| 113 | Undefined header |
| 120 | Numeric data error |
| 121 | Invalid character in numeric |
| 123 | Exponent too large |
| 124 | Too many digits |
| 130 | Suffix error |
| 131 | Invalid suffix |
| 134 | Suffix too long |

Table 3-5: Command Error Messages (CME Bit 5) (cont.)

| Code | Message |
|------|--------------------------|
| 140 | Character data error |
| 141 | Invalid character data |
| 144 | Character data too long |
| 150 | String data error |
| 151 | Invalid string data |
| 152 | String data too long |
| 160 | Block data error |
| 161 | Invalid block data |
| 170 | Command expression error |
| 171 | Invalid expression |

Execution Error The following table lists the execution errors that are detected during execution of a command.

Table 3-6: Execution Error Messages (EXE Bit 4)

| Code | Message |
|------|--|
| 200 | Execution error |
| 221 | Settings conflict |
| 222 | Data out of range |
| 224 | Illegal parameter value |
| 241 | Hardware missing |
| 250 | Mass storage error |
| 251 | Missing mass storage |
| 252 | Missing media |
| 253 | Corrupt media |
| 254 | Media full |
| 255 | Directory full |
| 256 | File name not found |
| 257 | File name error |
| 258 | Media protected |
| 259 | File name too long |
| 280 | Program error |
| 282 | Insufficient network printer information |
| 283 | Network printer not responding |
| 284 | Network printer server not responding |
| 286 | Program runtime error |

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

| Code | Message |
|------|---|
| 287 | Future use |
| 2200 | Measurement error, Measurement system error |
| 2201 | Measurement error, Zero period |
| 2202 | Measurement error, No period, second waveform |
| 2203 | Measurement error, No period, second waveform |
| 2204 | Measurement error, Low amplitude, second waveform |
| 2205 | Measurement error, Low amplitude, second waveform |
| 2206 | Measurement error, Invalid gate |
| 2207 | Measurement error, Measurement overflow |
| 2208 | Measurement error, No backwards Mid Ref crossing |
| 2209 | Measurement error, No second Mid Ref crossing |
| 2210 | Measurement error, No Mid Ref crossing, second waveform |
| 2211 | Measurement error, No backwards Mid Ref crossing |
| 2212 | Measurement error, No negative crossing |
| 2213 | Measurement error, No positive crossing |
| 2214 | Measurement error, No crossing, target waveform |
| 2215 | Measurement error, No crossing, second waveform |
| 2216 | Measurement error, No crossing, target waveform |
| 2217 | Measurement error, Constant waveform |
| 2219 | Measurement error, No valid edge - No arm sample |
| 2220 | Measurement error, No valid edge - No arm cross |
| 2221 | Measurement error, No valid edge - No trigger cross |
| 2222 | Measurement error, No valid edge - No second cross |
| 2223 | Measurement error, Waveform mismatch |
| 2224 | Measurement error, WAIT calculating |
| 2225 | Measurement error, No waveform to measure |
| 2226 | Measurement error, Null Waveform |
| 2227 | Measurement error, Positive and Negative Clipping |
| 2228 | Measurement error, Positive Clipping |
| 2229 | Measurement error, Negative Clipping |
| 2230 | Measurement error, High Ref < Low Ref |
| 2231 | Measurement error, No statistics available |
| 2233 | Requested waveform is temporarily unavailable |
| 2235 | Math error, invalid math description |
| 2240 | Invalid password |
| 2241 | Waveform requested is invalid |
| 2244 | Source waveform is not active |

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

| Code | Message |
|-------------|--|
| 2245 | Saveref error, selected channel is turned off |
| 2250 | Reference error, the reference waveform file is invalid |
| 2253 | Reference error, too many points received |
| 2254 | Reference error, too few points received |
| 2259 | File too big |
| 2270 | Alias error |
| 2271 | Alias syntax error |
| 2273 | Illegal alias label |
| 2276 | Alias expansion error |
| 2277 | Alias redefinition not allowed |
| 2278 | Alias header not found |
| 2285 | TekSecure(R) Pass |
| 2286 | TekSecure(R) Fail |
| 2500 | Setup error, file does not look like a setup file |
| 2501 | Setup warning, could not recall all values from external setup |
| 2620 | Mask error, too few points received |
| 2760 | Mark limit reached |
| 2761 | No mark present |
| 2762 | Search copy failed |

Device Error The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

Table 3-7: Device Error Messages (DDE Bit 3)

| Code | Message |
|-------------|-------------------------|
| 310 | System error |
| 311 | Memory error |
| 312 | PUD memory lost |
| 314 | Save/recall memory lost |

System Event The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

Table 3-8: System Event Messages

| Code | Message |
|-------------|--|
| 400 | Query event |
| 401 | Power on (PON bit 7 set) |
| 402 | Operation complete (OPC bit 0 set) |
| 403 | User request (URQ bit 6 set) |
| 404 | Power fail (DDE bit 3 set) |
| 405 | Request control |
| 410 | Query INTERRUPTED (QYE bit 2 set) |
| 420 | Query UNTERMINATED (QYE bit 2 set) |
| 430 | Query DEADLOCKED (QYE bit 2 set) |
| 440 | Query UNTERMINATED after indefinite response (QYE bit 2 set) |
| 468 | Knob/Keypad value changed |
| 472 | Application variable changed |

Execution Warning

The following table lists warning messages that do not interrupt the flow of command execution. They also notify you of possible unexpected results.

Table 3-9: Execution Warning Messages (EXE Bit 4)

| Code | Message |
|-------------|---|
| 528 | Parameter out of range |
| 532 | Curve data too long, Curve truncated |
| 533 | Curve error, Preamble values are inconsistent |
| 540 | Measurement warning, Uncertain edge |
| 541 | Measurement warning, Low signal amplitude |
| 542 | Measurement warning, Unstable histogram |
| 543 | Measurement warning, Low resolution |
| 544 | Measurement warning, Uncertain edge |
| 545 | Measurement warning, Invalid in minmax |
| 546 | Measurement warning, Need 3 edges |
| 547 | Measurement warning, Clipping positive/negative |
| 548 | Measurement warning, Clipping positive |
| 549 | Measurement warning, Clipping negative |

Table 3-10: Execution Warning Messages (EXE Bit 4)

| Code | Message |
|-------------|---|
| 540 | Measurement warning |
| 541 | Measurement warning, Low signal amplitude |

Table 3-10: Execution Warning Messages (EXE Bit 4) (cont.)

| Code | Message |
|------|---|
| 542 | Measurement warning, Unstable histogram |
| 543 | Measurement warning, Low resolution |
| 544 | Measurement warning, Uncertain edge |
| 545 | Measurement warning, Invalid min max |
| 546 | Measurement warning, Need 3 edges |
| 547 | Measurement warning, Clipping positive/negative |
| 548 | Measurement warning, Clipping positive |
| 549 | Measurement warning, Clipping negative |

Internal Warning

The following table shows internal errors that indicate an internal fault in the instrument.

Table 3-11: Internal Warning Messages

| Code | Message |
|------|--|
| 630 | Internal warning, 50 Ω overload |

Appendix A: Character Set

Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

| | | | |
|----------|---------------|--------------------|-------------------|
| *CAL | ACCM | AN | AUTOset |
| *CLS | ACCOMMONMODE | ANALOG | DPMAutoSet |
| *DDT | ACDC | ANALOG | AUTOset DPMPReset |
| *ESE | ACDCRMS | ANALYSISMethod | AUXLevel |
| *ESR | ACK | ANALYZemode | AUXiliary |
| *IDN | ACKMISS | AND | AUXout |
| *LRN | ACKnowledge | ANNOTate | AVAILable |
| *OPC | ACKnum | ANOISEDensity | AVERAGE |
| *OPT | ACQ | ANY | AVG |
| *PSC | ACQDURATION | ANYError | AVPosition |
| *PUD | ACquire | AOffset | AXPosition |
| *RST | ACquisition | APosition | AYPosition |
| *SRE | ACRMS | APPID | Auto |
| *STB | ACTION | APPPWR | B |
| *TRG | ACTONEvent | APPPWRSUM | B0 |
| *TST | ACTive | APPLY | B1 |
| *WAI | ADD | ARbitrary | B10 |
| 0 | ADDMEAS | AREA | B11 |
| 1 | ADDNew | AREA<t;x> | B12 |
| 1066 | ADDR10 | AREAofcrosssection | B13 |
| 1200 | ADDR7 | ARROW | B14 |
| 128 | ADDRANDDATA | ASCII | B15 |
| 1333 | ADDRESS | ASCIi | B16 |
| 1466 | ADDRESS | ASC_Fmt | B2 |
| 15 | ADDRESS | ASCIi | B3 |
| 1600 | ADVanced | ASIC | B4 |
| 1866 | AFG | ASOURCE | B5 |
| 2133 | AGND | ASPECTratio | B6 |
| 333 | AHPosition | ATRIGGER | B7 |
| 500E1 | ALARMSEARCh | AUDIO | B8 |
| 500E15 | ALERT | AUNIts | B9 |
| 500E3 | ALias | AUTHenticate | BACA |
| 7 | ALL | AUTO | BACKLight |
| 8 | ALLAcqs | AUTODim | BACKWard |
| 800 | ALLBits | AUTOETHERnet | BADDR |
| 9 | ALLEv | AUTOINCrement | BADGE |
| A | ALLTHResholds | AUTORange | BADge |
| AB | ALWAYS | AUTORbw | BANDwidth |
| ABC | AM14 | AUTOSAVEPITIMEOUT | BANDwidth |
| ABCB | AMPL | AUTOSAVEUITIMEOUT | BASE |
| ABORT | AMPLINear | AUTOScale | BASETop |
| ABOrt | AMPLITUDE | AUTOSet | BASic |
| ABSOLUTE | AMPLitude | AUTOZero | BATHTUB |
| ABSolute | AMPMode | AUTOfit | BATHtub |
| AC | AMP[1-10]val | AUTOMATIC | BC |
| ACBC | AMPITUDE | AUTOset | BCR |

| | | | |
|-----------------|-------------------|---------------------|-----------------|
| BCRType | BULK | CH4 | COMplete |
| BDIFFBP | BUNIts | CH5 | COMPosition |
| BEACon | BURSTDEtectmethod | CH6 | COMPression |
| BER | BURSTEDGTYPE | CH7 | CONDENRG |
| BHPOsition | BURSTLENGTH | CH8 | CONDLOSS |
| BIAS | BURSTLatency | CHAN1A | CONDition |
| BIN | BURSTWIDTH | CHAN2B | CONFIGuration |
| BINARY | BURSt | CHANwidth | CONFIGure |
| BINary | BUS | CHX | CONFigure |
| BIT | BUSANDWAVEFORM | Channe1 | CONNECTSTATUS |
| BITAMPLITUDE | BUSTABle | Checksum | CONNect |
| BITCfmode | BUSTURNAROUND | CH1t;x> | CONNECTed |
| BITDelay | BUSY | CH1t;x>_DALL | CONSTAmplitude |
| BITEnd | BUSx | CH1t;x>_D1t;x> | CONSTANTCLOCK |
| BITHIGH | BVPosition | CHx | CONSTCLOCKMODE |
| BITLOW | BXPosition | CHX_DALL | CONSTant |
| BITORder | BY | CHX_DX | CONTROL |
| BITOrder | BYPosition | CHX_Dy | CONTROLCHARTYPE |
| BITPERiod | BYT_Nr | CLASSA | CONTROLCODETYPE |
| BITPcnt | BYT_Or | CLASSB | CONTROLMESSAge |
| BITRate | BYTe | CLASSC | CONTRol |
| BITS | Block | CLASSD | CONTInuous |
| BITSTUFFing | B1t;x> | CLASS | CONTRol |
| BITStart | B1t;x>PSIFIVE | CLEAR | COPY |
| BITType | Bpeak | CLEAr | COUNT |
| BIT_Nr | Br | CLIPping | COUNTER |
| BIT1t;n>SOURCE | Bus | CLOCK | COUNT |
| BJT | Bx | CLOCKBITSPerCHANNEL | COUPLing |
| BLACKMANHarris | C | CLOCKFrequency | COUNT |
| BLACKmanharris | CA | CLOCKMultiplier | CPHY |
| BLANKingtime | CALLOUTS | CLOCKRecovery | CR |
| BLOCKDATA | CALLOUT1t;x> | CLOCKSource | CRc |
| BLOCKid | CALibrate | CLOCK | CRc16 |
| Blue | CAN | CLOCKSource | CRc5 |
| BM | CAN2X | CLRESPONSE | CRcError |
| BMP | CANH | CMETHOD | CRcHeader |
| BMSGEnd | CANL | CN | CRcTrailer |
| BN | CAPACITANCE | COFFSet | CREATOR |
| BNOISE | CARDIac | COLOR | CROSSIng |
| BN_Fmt | CATalog | COLOROFF | CSActive |
| BOFFSet | CB | COLORON | CSI |
| BOLD | CCODE | COLORy | CSIBLANK |
| BOOKMARK | CCOUNT | COLOR | CSINULL |
| BOOKMark | CCRESUlts | COLORs | CSLevel |
| BOTH | CCYClEs | COMM | CSMode |
| BPACKets | CENTERFREQUency | COMMAND | CSPLIT |
| BPOsition | CENTERFREQUency | COMMONMODE | CSSource |
| BRGTID | CGND | COMMONMode | CSYMLue |
| BRINGtoView | CGS | COMMand | CURRENT |
| BROADCASTPacket | CH | COMMENTS | CURRent |
| BROadcast | CH1 | COMPRESSEDPSStream | CURRentacq |
| BRSBit | CH2 | COMPRESSION | CURSOR |
| BSOURCE | CH3 | COMPensate | CURSOR1 |

| | | | |
|--------------------|---------------|----------------------|----------------|
| CURSOR<math>t;x> | DATA | DDT | DISPlaystat |
| CURSor | DBCA | DECIMAL | DISabled |
| CURSors | DBM | DECode | DISPlay |
| CURVe | DBMA | DEFect | DISPlaymode |
| CURrent | DBMV | DEFine | DIVide |
| CUSTOM | DBUA | DEFinedby | DIVisions |
| CUSTOMLIMITSFile | DBUV | DEGAUSS | DJ |
| CUSTOMPROPAGation | DBUW | DEGrees | DJDD |
| CUSTOMTABLe | DC | DELAY | DJDIRAC |
| CUSTOm | DCD | DELETE | DMINus |
| CWD | DCDC | DELETEALL | DMSGEnd |
| CYCLE | DCP1W2 | DELEte | DNS |
| CYCLEAmp | DCPWR | DELIMiter | DOFFSet |
| CYCLEBase | DCR | DELTA | DOMAINname |
| CYCLECount | DCREJ | DELTAtime | DOMain |
| CYCLEMAX | DCRType | DELTA | DONTInClude |
| CYCLEMin | DCSGLONGWRITE | DELTAtime | DONTcare |
| CYCLEMode | DCSLONGREAD | DELaY | DOTsonly |
| CYCLEPKPK | DCSR | DELeTe | DOUBLE |
| CYCLETop | DCSSRR | DELeTeqstring | DPHY |
| CYCLEcount | DCSSW | DELta | DPLUS |
| CYCLEmode | DDJ | DESE | DPMOVERSHOOT |
| ChpSelect | DDR | DESKEW | DPMRIPPLE |
| Chx_Dx | DDR3 | DESKew | DPMTURNOFFTIME |
| ClockEdge | DDRAOS | DEST | DPMTURNONTIME |
| Current | DDRAOSPERTCK | DESTination | DPMUNDERSHOOT |
| D | DDRAOSPERUI | DESTinationaddr | DPlus |
| DADDR | DDRAUS | DESTinationport | DQ0 |
| DAMPing | DDRAUSPERTCK | DEVErRor | DQDQS |
| DATA | DDRAUSPERUI | DEVICEChirp | DRIVE |
| DATA0 | DDRHOLDDIFF | DEVICEDESCMASTERREAD | DRive |
| DATA1 | DDRREAD | DEVICEDESCSLAVEREAD | DSI |
| DATA2 | DDRREADWRITE | DEVICETYPE | DSIBLANK |
| DATAA | DDRREADwrite | DEVICEType | DSINULL |
| DATAB | DDRRead | DEVicetype | Duty |
| DATABITS | DDRSETUPDIFF | DHCPbootp | DVBYDT |
| DATABITS | DDRTCHABS | DIAG | DVDT |
| DATABits | DDRTCHAVERAGE | DIBYDT | DVM |
| DATAFORMat | DDRTCKAVERAGE | DIDT | DWORD |
| DATAFORMat | DDRTCLABS | DIFF | DYNAMIC |
| DATAFormat | DDRTCLAVERAGE | DIFFerential | DYNAMIC |
| DATAMINUS | DDRTDQSK | DIGGR<math>t;x> | DYNRange |
| DATAMINUSTHRESHOLD | DDRTERRMN | DIR | DeLtaB |
| DATAMINUSTHRESHoId | DDRTERRN | DIRECTPacket | DeLtaH |
| DATAPLUS | DDRTJITCC | DIRacmodeI | D<math>t;x> |
| DATAPLUSTHRESHoId | DDRTJITDUTY | DIRect | EACHCLOCKCYCLE |
| DATAPacket | DDRTJITPER | DIRection | EBIT |
| DATAPath | DDRTPST | DISCMODE | EBITVALue |
| DATARATE | DDRTRPRE | DISCrete | ECC |
| DATARate | DDRTWPRE | DISLave | ECL |
| DATASource | DDRVIXAC | DISPLAY | ECU |
| DATATHRESHoId | DDRWRITE | DISPLAYUnits | ECUDATA |
| DATE | DDRwrite | DISParity | ECUSENSor |

| | | | |
|----------------|-------------------|--------------|-----------------|
| ECUSOURce | EQUation | EqualS | FILEName |
| EDECay | EQua1 | Execute | FILESYSTEM |
| EDGE | EQua1NOTEQua1 | F1MAG | FILESystem |
| EDGEInce | ERISe | F2 | FILTERRANGEFROM |
| EDGEQUALONE | ERROR | F3MAG | FILTERRANGETO |
| EDGEQUALTWO | ERRORREPort | F4 | FILTer |
| EDGEQUALifier | ERRORTYPE | F8 | FILTers |
| EDGES | ERRor | FACTory | FILEpath |
| EDGESOURce | ERRTYPE | FAIL | FIRST |
| EDGESource | ERRTYPE | FAILCount | FIVE |
| EDGEType | ERRType | FALL | FIVEHundred |
| EDGE<t;x> | ERRor | FALLHigh | FIXED |
| EDGE | ERRors | FALLING | FIXed |
| EFFICIENCY | ESCAPEMODE | FALLLow | FLATNESS |
| Efficiency | ESIBit | FALLMid | FLATTOP2 |
| EIGHTBIT | ET | FALLSLEWRATE | FLATtop2 |
| EIGHTHUNDREdHz | EVEN | FALLTIME | FLAG |
| EITHER | EVENT | FALLing | FLEXRAY |
| EITHer | EVENTBYTe | FALSe | FLEXray |
| EITHer | EVENTS | FALling | FLIP |
| EMBEDDED | EVENTtable | FAMILYCODE | FLYBACK |
| ENABle | EVERY | FAST | FOLDer |
| ENABle | EVMsg | FASTAcq | FONT |
| ENABled | EVQty | FASTERthan | FONTSize |
| ENCdg | EXCURsion | FASTer | FORCEDRange |
| ENCoding | EXECUTEQUEue | FASTframe | FORCe |
| END | EXECute | Fast | FORMERror |
| ENDPoint | EXITSwindow | FBD1 | FORMat |
| ENDSEARCh | EXPLICITCLOCK | FBD2 | FORward |
| ENDtime | EXPLICITCLOCKMODE | FBD3 | FOUR |
| ENET | EXPORTRaw | FC1063 | FOURBIT |
| ENET100 | EXTAtten | FC133 | FOURHUNDREDHZ |
| ENET1000 | EXTDBatten | FC2125 | FOURHUNDREdHz |
| ENgineering | EXTENDED | FC266 | FOURHundred |
| ENHANCED4 | EXTENDED | FC4250 | FOURTEENten |
| ENHANCED8 | EXTENDuis | FC531 | FP |
| ENSLave | EXTREGREAD | FC8500 | FPAnel |
| ENTERSwindow | EXTREGWRITE | FCS | FPBinary |
| ENTRDya | EXTUnits | FCSError | FRAME |
| ENTRTSTMode | EXTended | FD | FRAMECONTRol |
| ENTasx | EXTernal | FDBITS | FRAMEEND |
| ENvelope | EXTime | FDISO | FRAMEID |
| ENable | EYEDIAGRAM | FDNONISO | FRAMELENGth |
| EOF | EYEHIGH | FFT | FRAMEREJ |
| EOFTYPE | EYELow | FIBREchanne1 | FRAMESTART |
| EOP | EYEMASK | FIFTY | FRAMETYPe |
| EOPBITS | EYERENDER | FIFTYHZ | FRAMEType |
| EOT | EYERender | FIFTy | FRAMEType |
| EOTP | EYEWIDTHBER | FIFTyHz | FRAMING |
| EOTPDATa | EYehistogram | FIFTy | FRAME |
| EOW | Efficiency1 | FILE | FRAME |
| EOp | Efficiency2 | FILEDest | FREReference |
| Equal | Efficiency3 | FILEFormat | FREQ |

| | | | |
|----------------------|-----------------|------------------|-------------------|
| FREQBYTe | GM | HISTOGRAM | IMDAINPUTVOLTAGE |
| FREQUENCY | GMT | HIStory | IMDALINERIPPLE |
| FREQ[1-10]val | GPKNOB1 | HITDMVAlue | IMDAPHASORDIAGRAM |
| FREQ[1-11]val | GPKNOB2 | HITS | IMDAPLOTDisplay |
| FREQuency | GR | HITType | IMDAPOWERQUALITY |
| FROM | GRATICULE | HIVALue | IMDASWITCHRIPPLE |
| FROMEDGESEARCHDIRect | GRATicule | HOLD | IMDATIMETREND |
| FROMLevel | GRAticule | HOLDTime | IMPEDANCE |
| FROMSearch | GREATERthan | HOLDoff | IMPEDance |
| FROMSymbol | GREen | HOMEDir | IN |
| FROMedge | GRIDTYPE | HORDER | INACCBYTe |
| FRame | GRIDlines | HORIZontalscale | INCLUDEREFs |
| FULL | GRId | HORZ | INCLUDE |
| FULLSPEED | GROUPDelay | HORZPOS | INCLUDE |
| FULLScreen | GSLSb | HORZScale | INDBits |
| FULl | GSMSb | HORizonta1 | INDEPENDENT |
| FUNCTIONCODETYPE | GSP | HOSTChirp | INDUCT |
| FUNCTIONCODE | GSRR | HOTJoin | INDUCTANCE |
| FUNction | GSW | HS | INDEX |
| FUNDCURrent | H | HSOURce | INFILTErs |
| Fw1394BS1600B | HAMming | HSYNCEND | INFinite |
| Fw1394BS400B | HANDSHAKEPacket | HSYNCSTART | INFMAXimum |
| Fw1394BS800B | HANDSHAKEType | HUB | INFMINimum |
| False | HANNing | HUNDBASETx | INFPersist |
| GAIN | HARMONICS | HUNDred | INFormation |
| GAINCROSSOVERFREQ | HARMONICSSClass | HYSTEResis | INIT |
| GAP | HARMONICSSOURce | HYSTEResis | INPUT |
| GATESource | HARMONICSStd | Hc | INPUTCAP |
| GATEWay | HARMONICSSUNits | Hmax | INPUTLeve1 |
| GATing | HAVERSINE | I2C | INPUTMode |
| GAUSSian | HBars | I2S | INPUTPower |
| GAUSSian | HDRCapability | IBA2500 | INPUTSource |
| GENIPADDRESS | HDRExit | IBA_GEN2 | INPUTType |
| GENERator | HDRRestart | ICFACTOR | INPWR |
| GET | HEADER | ID | INPWRSUM |
| GETBusch | HEADer | IDANDDATA | INPut |
| GETDevch | HEIGHT | IDFormat | INRUSH |
| GETMRd1 | HEIGHTBER | IDLE | INRUSHcurrent |
| GETMwr1 | HEIGHT | IDLEtime | INSIDEGreater |
| GETPrid | HEX | IDLe | INSIDERange |
| GETReg | HEXAgon | IDentifier | INSTR |
| GETRegpkta1ert | HFRej | IEC | INSTAll |
| GETRegpkbtad | HI | IEEE519 | INTENSITY |
| GETRegpktrecent | HID | ILEVELAbs | INTERNa1 |
| GETRegtestcfg | HIGH | ILEVELPct | INTERPRatio |
| GETRegvndor | HIGHLeve1 | IMAGE | INTERRUPT |
| GETRegvrent | HIGHLevel | IMAGinary | INTERNa1 |
| GETSlave | HIGHLimit | IMDAACQTREND | INTERp |
| GLoBa1 | HIGHPass | IMDADQ0 | INTerna1 |
| GLoBa1ref | HIGHREFVOLTage | IMDAEFFICIENCY | INVERTColor |
| GLONGREAD | HIGHTIME | IMDAHARMONICS | INVERTNIBble |
| GLONGWRITE | HIGHZ | IMDAINPUTCURRENT | INVERTed |
| GLoBa1 | HIRes | IMDAINPUTPOWER | INVERTed |

| | | | |
|---------------|--------------------|-------------------|-------------------|
| INVert | LESSthan | LOWLevel | MATHFFTx |
| INVerted | LEVELUNIts | LOWLimit | MATH<lt;x> |
| INputs | LEVel | LOWPass | MATHx |
| INrange | LF | LOWREFVoItage | MATlab |
| IO | LFREQuency | LOWSPEED | MAX |
| IOUT1SOURCE | LFRej | LOWTHRESHoId | MAXCAS |
| IOUT2SOURCE | LIC | LOWTIME | MAXCycLe |
| IOUT3SOURCE | LICENSE | LOWer | MAXFRames |
| IPADDRESS | LICense | LOWerthreshold | MAXHoId |
| IPHeader | LIMIT | LP | MAXIMUM |
| IPOWER | LIMITCount | LPDDR3 | MAXPSRR |
| IPVFOUR | LIMit | LPDT | MAXPSRRFREQ |
| IRMS | LIN | LPS666 | MAXREAD |
| IRipple | LINE | LPTHRESHoId | MAXRETSIZE |
| ISCLOCKED | LINEAR | LPYCBCR24 | MAXRTURN |
| ISOALL | LINEAr | LRIPPKPK | MAXSamplerate |
| ISOEND | LINEEND | LRIPRMS | MAXTIME |
| ISOMID | LINEFREQUency | LSB | MAXVoItage |
| ISOSTART | LINERIPPLE | LSlave | MAXWRITE |
| ISOURCE | LINESTART | LXI | MAXimum |
| ISOURCE | LINESelected | LogicState | MDATA |
| ISOchronous | LINear | MAC | MDATASpeed |
| ITALIC | LIST | MACADDRESS | ME |
| ITALic | LJ | MACLEnGth | MEAN |
| ITEM | LOAD | MAG | MEANAUTOCalculate |
| IVSINTEGRALV | LOCKCenter | MAGLOSS | MEANhistogram |
| IVSINTV | LOCKRJ | MAGNETICLOSS | MEAS |
| InputPwr | LOCKRJValue | MAGNITUDE | MEAS1 |
| Inrange | LOCKSpEctrum | MAGNitude | MEASRange |
| J2 | LOCK | MAGPROPERTY | MEASTABle |
| J9 | LOG | MAINWindow | MEASU |
| JITTERMODE | LOGARITHM | MAIn | MEASUREAT |
| JITTERSUMMARY | LOGIC | MANChester | MEASurement |
| JITTERSummary | LOGIC1SOURCE | MANUAL | MEAS<lt;x> |
| JITTERmodel | LOGIC2SOURCE | MANuaI | MEASured |
| JPG | LOGIC3SOURCE | MARKER | MEASurement |
| JTFBandwidth | LOGIC4SOURCE | MARgin | MEASx |
| KAISERBessel | LOGIC5SOURCE | MASK | MEDian |
| KAISerbessel | LOGICPattern | MASKFail | MEDium |
| L | LOGICQUALification | MASKHit | MEMORY |
| L2LTOL2N | LOGICSource | MASKOffset | MESSaging |
| LABELANDDATA | LOGIC | MASKPass | METHoD |
| LABel | LOGic | MASKfile | MID |
| LAN | LONG | MASK<lt;x> | MIDRef |
| LATCH | LONGEXTREGREAD | MASTERADDRESS | MIL |
| LAYOUT | LONGEXTREGWRITE | MASTERREAD | MIN |
| LDIR | LOOP | MASTERWRITE | MINCas |
| LEFt | LOOPBandwidth | MASTer | MINCycLe |
| LENGth | LOREntz | MATCHROM | MINHoId |
| LENGth | LOW | MATH | MINIMUM |
| LESSEQUAL | LOWDATAMINus | MATHArbfl<lt;x> | MINimum |
| LESSLimit | LOWDATAPLUS | MATHFFTView<lt;n> | MINMax |
| LESSThan | LOWERFREQUency | MATHFFTView<lt;x> | MINPSRR |

| | | | |
|------------------|-------------------|-------------------|-------------------|
| MINPSRRFREQ | NOISerej | OCCURS | OUTPUT7SOURCE |
| MINUI | NOISE | ODD | OUTPUT7VOLTage |
| MINimum | NOMINALOFFset | ODDEven | OUTPUTSource |
| MISO | NOMinal | OFF | OUTPUTType |
| MIXED | NOMinalfreq | OFF1 | OUTPUTt;n>VOLTage |
| MIXEDASCII | NONE | OFFSet | OUTPWR |
| MIXEDHEX | NONTRANSition | ON | OUTPWRSUM |
| MIXed | NOne | ONCE | OUTPut |
| MKDir | NOPARity | ONE | OUTSIDEgreater |
| MODE | NOR | ONEOVERDELTAVALUE | OUTSIDErange |
| MODEhistogram | NORESPonse | ONEPAIRI | OUTSOURCE |
| MODE | NORMALColor | ONEPAIRV | OUTWIRing |
| MODEl | NORMAL | ONEPAIRVI | OUTrange |
| MODulo | NORMAL | ONEWIRE | OUTside |
| MOREEQua | NOSTATION | ONFAIL | OVERDRIVEMATCH |
| MOREEQua1 | NOTE | OPCODEERRor | OVERDRIVESKIP |
| MOREEQua1INrange | NOTEQua1s | OPCode | OVERDRIVE |
| MORELimit | NOVERSHOOT | OPERating | OVERLoad |
| MOREThan | NOne | OPPOSITEas | OVERride |
| MOREthan | NPERIOD | OPTIMIZatiON | OVERlay |
| MOSFET | NPJ | OPTIMize | OVERdrive |
| MOSi | NPULSE | OPTIONALPARAm | OVERlay |
| MOUNT | NPULSEWIDTH | OPTion | Output1Pwr |
| MOVEABLE | NR1 | OPTional | Output2Pwr |
| MSB | NR2 | OR | Output3Pwr |
| MULTiply | NR3 | OUT | P1W2V1I1 |
| MULTipleframes | NRZ | OUTEDGEQUALifier | P1W3V2I2 |
| Manual | NR_Pt | OUTFILTERs | P3W3 |
| Mathlt;x> | NTIMES | OUTL2LTOL2N | P3W3V2I2 |
| Mathx | NTIMes | OUTLINESelected | P3W3V3I3 |
| N | NULL | OUTPUT | P3W4 |
| NACK | NULLFRDynamic | OUTPUT1 | PACKET |
| NAK | NULLFRStatic | OUTPUT1SOURCE | PACKets |
| NAME | NUL1 | OUTPUT1Type | PALETTE |
| NAME | NUMACQs | OUTPUT1VOLTage | PANKNOB |
| NAND | NUMACq | OUTPUT2 | PARityerror |
| NAND | NUMAVg | OUTPUT2SOURCE | PARallel |
| NATIVE | NUMBER | OUTPUT2Type | PARity |
| NAVigate | NUMBins | OUTPUT2VOLTage | PASSFAILENabled |
| NDUTY | NUMCHANnel | OUTPUT3 | PASSFAILHIGHlimit |
| NDUTYCYCLE | NUMERICORder | OUTPUT3SOURCE | PASSFAILLIMit |
| NDUTY | NUMFRAMESACQuired | OUTPUT3Type | PASSFAILLOWlimit |
| NDuty | NUMOFOutputs | OUTPUT3VOLTage | PASSFAILMARGIn |
| NEGative | NUMOUTputs | OUTPUT4 | PASSFAILWHEN |
| NEGative | NUMSEQUence | OUTPUT4SOURCE | PASSWord |
| NETWORKCONFig | NWIDTH | OUTPUT4VOLTage | PATTERNDETECTION |
| NEWpass | NWIDTTH | OUTPUT5 | PATTERNLEngth |
| NEXT | NYET | OUTPUT5SOURCE | PATTERNType |
| NIBBLECount | None | OUTPUT5VOLTage | PATtern |
| NIBBLE | OC1 | OUTPUT6 | PAUSE |
| NO | OC12 | OUTPUT6SOURCE | PAUSEPULSe |
| NOCARE | OC3 | OUTPUT6VOLTage | PAUSE |
| NOISEAdd | OC48 | OUTPUT7 | PAYLEngth |

| | | | |
|--------------------|-------------------|----------------|--------------|
| PAYLOAD | PNG | PSRR | RATE33K |
| PAYLength | PNJUNCTION | PTYPE | RATE38K |
| PAYLoad | POHCL | PT_Fmt | RATE3M |
| PAYload | POHCM | PT_Off | RATE400K |
| PCIE_GEN1 | POHCS | PULSEWIDTH | RATE4K |
| PCIE_GEN2 | POINT | PULSEwidth | RATE4M |
| PCIE_GEN3 | POINTS | PULSe | RATE500K |
| PCIExpress | POLARITY | PULSe | RATE50K |
| PDUTTY | POLarity | PWIDTH | RATE5M |
| PDUTY | POPulation | PWRFACTOR | RATE62K |
| PDUTYCYCLE | PORT | PWRFREQ | RATE68K |
| PEAK | PORTConfiguration | PWRUpstatus | RATE6M |
| PEAKCURRENT | PORTReset | ParalBlockgt | RATE7M |
| PEAKVOLTage | POSITIVE | Permeability | RATE800K |
| PEAKdetect | POST | QFACTOR | RATE83K |
| PERCent | POSTAMBLE | QString | RATE8M |
| PERFREQ | Position | QStringNR1 | RATE921K |
| PERIOD | Positive | QStringQString | RATE92K |
| PERIod | POVERSHOOT | QTAG | RATE9K |
| PERSistence | POWERFACTOR | QTAGGING | RATE9M |
| PERSource | POWERQUALITY | QUALifier | RATed |
| PFACTOR | POWERRating | QUALifier | RAW10 |
| PFC | POWERx | Qstring | RAW12 |
| PHASE | POWLINear | RADians | RAW14 |
| PHASECROSSOVERFREQ | POWLOG | RAIL17 | RAW16 |
| PHASENOISE | Power | RAILNUM | RAW20 |
| PHASEONE | Power<lt;x> | RAMP | RAW6 |
| PHASETHREE | PPD | RAMPtime | RAW7 |
| PHASETWO | PPS101010 | RANDom | RAW8 |
| PHASOR | PPS121212 | RAP | RBW |
| PHASE | PPS565 | RAPDATA | RBWMode |
| PHYSICALADDRESS | PPS666 | RATE100K | RCURRent |
| PHYSICALaddress | PPS888 | RATE10K | RDS |
| PICPARAMeter | PPULSE | RATE10M | RDSOn |
| PID | PPULSEWIDTH | RATE115K | RDSON |
| PING | PREAMBLE | RATE11M | RDSON |
| PIXELNUMBer | PRESET | RATE125K | READ |
| PIXELVALue | PRESS | RATE12M | READFile |
| PIXel | PREViewstate | RATE13M | READROM |
| PJ | PREvious | RATE14M | READOut |
| PK2PK | PREamble | RATE153K | READY |
| PK2Pk | PREsence | RATE15M | REAL |
| PKPK | PRIMARYTURNS | RATE16M | RECA11 |
| PLL | PRIOD | RATE19K | RECA11mask |
| PLOT | PROBEControl | RATE1K | RECORD |
| PLOTData | PROBECa1 | RATE1M | RECORDLength |
| PLOTVIEW<lt;x> | PROBEFunc | RATE20K | RECOrdLength |
| PLOTVIEWx | PROFile | RATE250K | RECTANGLE |
| PLOTview<lt;n> | PROToCol | RATE25K | RECTANGular |
| PLOTview<lt;x> | PROToCol | RATE2K | RECTangle |
| PLOT<lt;x> | PROToCol | RATE2M | RECTangular |
| PLOTx | PRObe | RATE300 | RED |
| PM | PSIFIVE | RATE31K | REF |
| | | | REFERence |

| | | | |
|--------------------|------------------|-----------------|-----------------|
| REFFFTView<n> | RESPonsetime | RRBDisplaystate | SEARCH |
| REFFFTView<x> | RESUME | RS232C | SEARCH1 |
| REFFFTx | RESUlt | RSTDya | SEARCHOPTION |
| REFLEVELMode | RESUlts | RUN | SEARCHROM |
| REFLevel | RESAmple | RUNSTop | SEARCHSource |
| REFLevels | RESet | RUNT | SEARCHTABle |
| REFLevels<x> | RESistance | RUNt | SEARCHTable |
| REFMode | RF_AVERage | RWINADDR | SEARCH<x> |
| REFOUT | RF_FREQuency | RX | SEARCHtotrigger |
| REFVOLTAGE<n>Val | RF_FREQuency<x> | S8B10B | SEARCHx> |
| REFVoltage | RF_MAGNitude | SADDRESS | SEARCH |
| REFERENCE | RF_MAGNitude<x> | SAME | SEC1SOURCE |
| REF<x> | RF_MAXHold | SAMEas | SEC1TURNS |
| REF<x>_DALL | RF_MINHold | SAMPLERate | SEC2SOURCE |
| REF<x>_D<x> | RF_NORMAl | SAMPLEpoint | SEC2TURNS |
| REFx | RF_PHASE | SAMple | SEC3SOURCE |
| REFx_DX | RF_PHASE<x> | SAS12_NOSSC | SEC3TURNS |
| REGOWRite | RFvsTime | SAS12_SSC | SEC4SOURCE |
| REGAddrtestcfg | RGB444 | SAS15_NOSSC | SEC4TURNS |
| REGAddrvndor | RGB555 | SAS15_SSC | SEC5SOURCE |
| REGDatatestcfg | RGB565 | SAS3_NOSSC | SEC5TURNS |
| REGDatavndor | RGB666 | SAS3_SSC | SEC6SOURCE |
| REGISTERADDRESS | RGB888 | SAS6_NOSSC | SEC6TURNS |
| REGREAD | RI | SAS6_SSC | SECOND |
| REGWRite | RIBinary | SATA_GEN1 | SECONds |
| REGion | RIGHt | SATA_GEN2 | SECPhase |
| REGister | RI0125 | SATA_GEN3 | SECVo<t |
| REGisteraddress | RI0250 | SAVEIMAGE | SECWINDings |
| REJ | RI03125 | SAVEON | SECONds |
| REJect | RIPPLEFREQ<n>Val | SAVEONEEvent | SEG<x> |
| REM | RISE | SAVEWAVEform | SEG<x>COUNT |
| REMOte | RISEHigh | SAVE | SEG<y> |
| REName | RISELow | SAVEmask | SELECTED |
| REPEATERHOST | RISEMId | SC | SELECTIONtype |
| REPEATERPERIPHERAL | RISESLEWRATE | SCALE | SELTrace |
| REPEATStart | RISETIME | SCALERATio | SELEct |
| REPEATstart | RISING | SCALE | SELEcted |
| REPOrt | RISe | SCALE | SENSORADDRESS |
| REPWR | RISing | SCIENTific | SENSORECU |
| REPWRSUM | RJ | SCLk | SENSORSTATUS |
| REPeating | RJDD | SCOPEApp REBOOT | SENsor |
| REQDISConnect | RJDIRAC | SCRAMBLING | SENT |
| REQSETINIT | RMDir | SCREEN | SEQSETup |
| REQUested | RMS | SCRambling | SEQnum |
| RERUN | RNR | SDATA | SEQUence |
| RESAMPLE | ROLL | SDI | SERIALNUMBer |
| RESAMPLERate | ROLOCATION | SDIDATA | SERVICE |
| RESERVED | ROMCODE | SDIDATASSM | SERVICEMode |
| RESET | ROSc | SDLC | SERVICENAME |
| RESETTRIGger | ROtation | SDR | SERnumber |
| RESOLution | RP | SDRBroadcast | SESSion |
| RESPONSECODE | RPBinary | SDRDirect | SET |
| RESPonse | RR | SE | SETBr<t |

| | | | |
|-----------------|----------------|----------------|-----------------|
| SETDecay | SLAVE | SRJ | Standard |
| SETDya | SLEEP | SRPbinary | STop |
| SETFast | SLEWRATEMethod | SRQ | SUBADdress |
| SETHOLDLogicval | SLEep | SS | SUBGROUP |
| SETHold | SLOW | SSC | SUBNETMask |
| SETLevel | SLOWERthan | SSCFREQDEV | SUBSF |
| SETMrdl | SLOWER | SSCMODRATE | SUBTYpe |
| SETMwr1 | SLOpe | SSCPROFILE | SUBtract |
| SETNdya | SMPS | SSM | SUMFrame |
| SETPs | SNap | SSPLIT | SUPPORTBYTe |
| SETRegaddr | SNRM | STADdress | SUPPress |
| SETRegdata | SNRME | STALL | SUPervisory |
| SETSlow | SOA | STANDARD | SUSPEND |
| SETTime | SOAHITSCNT | STANDARD | SV |
| SETUP | SOCKETServer | START | SWITCHING |
| SETUp | SOF | STARTBIT | SWITCHINGLOSS |
| SETWp | SOFFRAMENUMber | STARTCONDition | SWITCHINGRIPPLE |
| SEText | SOT | STARTFREQUency | SWL |
| SEVEN | SOURCE | STARTFREQUency | SWLCONFIgType |
| SFD | SOURCEEDGEType | STARTFrequency | SWRIPPkPK |
| SFPbinary | SOURCEList | STARTPACKet | SWRIPRMS |
| SHAPE | SOURCETyPe | STARTOffframe | SYMB |
| SHORT | SOURCES | STARTtime | SYMBol |
| SHOWAREas | SOURCE | STARTup | SYMBol |
| SHOWCRiteria | SOURce | STARTupnosync | SYMBOLs |
| SHOwEquation | SOURceaddr | START | SYMMetry |
| SHort | SOURcelt;x> | STATE | SYNC |
| SHUTDOWN | SOURceport | STATEBYTe | SYNCBITS |
| SHUTdown | SOW | STATIC | SYNCFrame |
| SI | SPACEWIRE | STATIONADDR | SYNCMODE |
| SIGNAL | SPAN | STATIStics | SYNCSEARCh |
| SIGNALFREQUency | SPANABovebw | STATUS | SYNCTHRESHold |
| SIGNALType | SPANBELOWdc | STATE | SYNCfield |
| SIGNALType | SPANRBWRatio | STATIC | SYSTEM |
| SIGNal | SPEC | STATUS | Start |
| Signal | SPECIALPacket | STAYSHigh | Stop |
| SINC | SPECIALType | STAYSLow | String |
| SINE | SPECTRAL | STACked | Symbol |
| SINGLE | SPECTRALBUJ | STANDard | TARGETBER |
| SINX | SPECTRUM | STDDev | TBIT |
| SIX | SPECTral | STEPRESPONSE | TBITRDNeg |
| SIXBIT | SPECViewlt;x> | STOP | TBITRDPos |
| SIXFIFTyhz | SPECTral | STOPACQ | TBITVAlue |
| SIXTEENeight | SPECTrum | STOPACq | TCKAVG |
| SIXTYHZ | SPI | STOPAAfter | TCPHeader |
| SIXTYhz | SPLIT | STOPFREQUency | TDM |
| SIXty | SPLITMODE | STOPFrequency | TDMVAlue |
| SIZE | SPLit | STROBE | TEKEXponential |
| Size | SPREADSheet | STROBESource | TEKSecure |
| SKEW | SPace | STRObe | TEMPerature |
| SKIPROM | SQUELCH | STRObe | TENBASET |
| SLAVE | SQUare | STYpe | TENNinety |
| SLAVEADDRESS | SRIbinary | STYle | TERMinal |

| | | | |
|--------------------|----------------------|----------------|---------------|
| TERmination | TRACK | UNIQUE | VOUT1SOURCE |
| TEST | TRAILER | UNITINTERVAL | VOUT2SOURCE |
| TESTCONNECTION | TRANSFERBUSOWNERSHIP | UNITINTERVALS | VOUT3SOURCE |
| TESTMODE | TRANSITION | UNITINTERVALS | VRMS |
| TESTmode | TRANSITION | UNITS | VSOURCE |
| TEST | TRANstion | UNITs | VSOURCE |
| TEXT | TRAPezoid | UNLock | VSYNCEnd |
| TF | TRAIler | UNLocked | VSYNcSTART |
| THDF | TRBit | UNMOUNT | VTERM |
| THDR | TRIAngle | UNNumbered | VUNIT |
| THIRD | TRIGLevel | UNWRap | WAITTime |
| THREE | TRIGger | UP | WAKEup |
| THREEPHASEAUTOset | TRIGgertosearch<! | UPDatenow | WAKEup |
| THREESIXtyhz | TRPWR | UPPERFREQUENCY | WAVEFORM |
| THRESHold | TRPWSUM | UPPer | WAVEVIEW1 |
| THReshold | TRUEPWR | UPPerthreshold | WAVEview |
| TICKTIME | TRUe | USAGE | WAVEview1t;x> |
| TICKTOLerance | TRack | USB3 | WAVEview1t;y> |
| TICKS | TTL | USBDevice | WAVEform |
| TIE | TTLENRG | USBTmc | WAVEforms |
| TIEHISTOGRAM | TTLLOSS | USECLockedge | WAVFrm |
| TIESPECTRUM | TURN | USECLockedge | WEIGHT |
| TIETIMETREND | TURNOfftime | USEGLOBAL | WFId |
| TIMECode | TURNON | USEREIGHTB | WFMOutpre |
| TIMEOUTSIDELEVEL | TURNONTIME | UTCDELTA | WFMTYPE |
| TIMEOut | TURNONtime | V | WHEN |
| TIMESamp | TWELVetwelve | V1X | WIDTH |
| TIMETOMAX | TWENTYBIT | V2X | WIDTHBER |
| TIMETOMIN | TWENTyeighty | VALUe | WIDTh |
| TIMETREND | TWENTyfive | VALidate | WINDOW |
| TIMINGMode | TWO | VALUe | WINDOWLENGth |
| TIME | TWOBIT | VARpersist | WINSCALE |
| TJBER | TWOCLOCKCYCLES | VBArs | WINDow |
| TNRATIO | TWOFifty | VCESat | WIRing |
| TO | TWOTHousand | VCFACTOR | WIThin |
| TOEDGESEARCHDIRect | TX | VDIFFXOVR | WORD |
| TOEdge | TXRX | VECTors | WORDCOUNT |
| TOFFENRG | TXRXTHRESHold | VERBoSe | WORDSIZE |
| TOFFLOSS | TYPE | VERTICES | WORDSel |
| TOKENPacket | TYPE1 | VERTPOS | WORDSize |
| TOKENType | TYPE2 | VERTSCALE | WRAP |
| TOLERance | TYPe | VERTical | WRITE |
| TOLerance | TotalEfficiency | VERsion | WRITEFile |
| TOLevel | True | VGLLevel | WRITE |
| TONENRG | U | VIEW | Word |
| TONLOSS | UACK | VIEWStyle | X |
| TOOTHGAP | UI | VIEWTYpe | XAUI |
| TOP | ULPS | VISIBility | XAUI_GEN2 |
| TOSYmbol | UNDERLine | VISuaL | XAXIS |
| TOTAL | UNDERline | VLEVELAbs | XAXISunits |
| TOTALEFFiciency | UNDO | VLEVELPct | XFF |
| TOTALuptime | UNEQual | VOLTAGE | XID |
| TOUCHSCREEN | UNINSTALL | VOLTage | XINcr |

Appendix C: Factory Defaults

Default Setup

The following table lists the default values for each command.

NOTE. Find the most up-to-date default values for your instrument and software by performing a TekSecure command, saving the instrument setup and looking at the instrument or setup file.

Table C-1: Default Values

| Item | Description |
|---|--------------|
| ACQUIRE:FASTACQ:PALETTE | TEMPERATURE |
| ACQUIRE:MAGNIVU | 0 |
| ACQUIRE:MODE | SAMPLE |
| ACQUIRE:NUMAVG | 16 |
| ACQUIRE:NUMENV | INFINITE |
| ACQUIRE:STATE | 1 |
| ACQUIRE:STOPAFTER | RUNSTOP |
| ACTONEVENT:ACTION:AUXOUT:STATE | 0 |
| ACTONEVENT:ACTION:EMAIL:SETUP:TOADDRESS | "TestString" |
| ACTONEVENT:ACTION:EMAIL:STATE | 0 |
| ACTONEVENT:ACTION:PRINT:STATE | 0 |
| ACTONEVENT:ACTION:SAVEIMAGE:STATE | 0 |
| ACTONEVENT:ACTION:SAVEWFM:STATE | 0 |
| ACTONEVENT:ACTION:SRQ:STATE | 0 |
| ACTONEVENT:ACTION:STOPACQ:STATE | 1 |
| ACTONEVENT:ACTION:VISUAL:STATE | 0 |
| ACTONEVENT:EVENTTYPE | NONE |
| ACTONEVENT:NUMACQS | 1 |
| ACTONEVENT:REPEATCOUNT | 1.0000 |
| AFG:AMPLITUDE | 500.0000E-3 |
| AFG:ARBITRARY:EMEM:POINTS:ENCDG | ASCII |
| AFG:FREQUENCY | 100.0E+3 |
| AFG:FUNCTION | SINE |
| AFG:HIGLEVEL | 250.0000E-3 |
| AFG:LEVELPRESET | USER |
| AFG:LOWLEVEL | -250.0000E-3 |
| AFG:NOISEADD:PERCENT | 0.0E+0 |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------------------|-------------------|
| AFG:NOISEADD:STATE | 0 |
| AFG:OFFSET | 0.0E+0 |
| AFG:OUTPUT:LOAD:IMPEDANCE | HIGHZ |
| AFG:PERIOD | 9.999999999916E-6 |
| AFG:PHASE | 0.0E+0 |
| AFG:PULSE:WIDTH | 1.0E-6 |
| AFG:RAMP:SYMMETRY | 50.0000 |
| AFG:SQUARE:DUTY | 50.0000 |
| ALIAS:STATE | 0 |
| APPLICATION:TYPE | POWER |
| AUXOUT:EDGE | RISING |
| AUXOUT:SOURCE | ATRIGGER |
| BUS:B1:CAN:BITRATE | 500000 |
| BUS:B1:CAN:FD:BITRATE | 4000000 |
| BUS:B1:CAN:FD:STANDARD | ISO |
| BUS:B1:CAN:SAMPLEPOINT | 50 |
| BUS:B1:CAN:SOURCE | CH1 |
| BUS:B1:CAN:STANDARD | CAN2X |
| BUS:B1:DISPLAY:FORMAT | HEXADECIMAL |
| BUS:B1:DISPLAY:TYPE | BUS |
| BUS:B1:I2C:ADDRESS:RWINCLUDE | 0 |
| BUS:B1:I2C:CLOCK:SOURCE | CH1 |
| BUS:B1:I2C:DATA:SOURCE | CH2 |
| BUS:B1:LABEL | "Parallel" |
| BUS:B1:LIN:BITRATE | 19200 |
| BUS:B1:LIN:IDFORMAT | NOPARITY |
| BUS:B1:LIN:POLARITY | NORMAL |
| BUS:B1:LIN:SAMPLEPOINT | 50 |
| BUS:B1:LIN:SOURCE | CH1 |
| BUS:B1:LIN:STANDARD | V2X |
| BUS:B1:PARALLEL:BIT0:SOURCE | D0 |
| BUS:B1:PARALLEL:BIT10:SOURCE | D10 |
| BUS:B1:PARALLEL:BIT11:SOURCE | D11 |
| BUS:B1:PARALLEL:BIT12:SOURCE | D12 |
| BUS:B1:PARALLEL:BIT13:SOURCE | D13 |
| BUS:B1:PARALLEL:BIT14:SOURCE | D14 |
| BUS:B1:PARALLEL:BIT15:SOURCE | D15 |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------------------------|-------------|
| BUS:B1:PARALLEL:BIT16:SOURCE | CH1 |
| BUS:B1:PARALLEL:BIT17:SOURCE | CH2 |
| BUS:B1:PARALLEL:BIT18:SOURCE | CH3 |
| BUS:B1:PARALLEL:BIT19:SOURCE | CH4 |
| BUS:B1:PARALLEL:BIT1:SOURCE | D1 |
| BUS:B1:PARALLEL:BIT2:SOURCE | D2 |
| BUS:B1:PARALLEL:BIT3:SOURCE | D3 |
| BUS:B1:PARALLEL:BIT4:SOURCE | D4 |
| BUS:B1:PARALLEL:BIT5:SOURCE | D5 |
| BUS:B1:PARALLEL:BIT6:SOURCE | D6 |
| BUS:B1:PARALLEL:BIT7:SOURCE | D7 |
| BUS:B1:PARALLEL:BIT8:SOURCE | D8 |
| BUS:B1:PARALLEL:BIT9:SOURCE | D9 |
| BUS:B1:PARALLEL:CLOCK:EDGE | RISING |
| BUS:B1:PARALLEL:CLOCK:ISCLOCKED | NO |
| BUS:B1:PARALLEL:CLOCK:SOURCE | CH1 |
| BUS:B1:PARALLEL:WIDTH | 16 |
| BUS:B1:POSITION | 0.0E+0 |
| BUS:B1:RS232C:BITRATE | 9600 |
| BUS:B1:RS232C:DATABITS | 8 |
| BUS:B1:RS232C:DELIMITER | LF |
| BUS:B1:RS232C:DISPLAYMODE | FRAME |
| BUS:B1:RS232C:PARITY | NONE |
| BUS:B1:RS232C:POLARITY | NORMAL |
| BUS:B1:RS232C:RX:SOURCE | OFF |
| BUS:B1:RS232C:TX:SOURCE | CH1 |
| BUS:B1:SPI:BITORDER | MSB |
| BUS:B1:SPI:CLOCK:POLARITY | RISE |
| BUS:B1:SPI:CLOCK:SOURCE | CH1 |
| BUS:B1:SPI:DATA:IN:POLARITY | HIGH |
| BUS:B1:SPI:DATA:IN:SOURCE | OFF |
| BUS:B1:SPI:DATA:OUT:POLARITY | HIGH |
| BUS:B1:SPI:DATA:OUT:SOURCE | CH3 |
| BUS:B1:SPI:DATA:SIZE | 8 |
| BUS:B1:SPI:FRAMING | SS |
| BUS:B1:SPI:IDLETIME | 5.0000E-6 |
| BUS:B1:SPI:SELECT:POLARITY | LOW |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------------------|-------------|
| BUS:B1:SPI:SELECT:SOURCE | CH2 |
| BUS:B1:STATE | 0 |
| BUS:B1:TYPE | PARALLEL |
| BUS:B2:CAN:BITRATE | 500000 |
| BUS:B2:CAN:FD:BITRATE | 4000000 |
| BUS:B2:CAN:FD:STANDARD | ISO |
| BUS:B2:CAN:SAMPLEPOINT | 50 |
| BUS:B2:CAN:SOURCE | CH1 |
| BUS:B2:CAN:STANDARD | CAN2X |
| BUS:B2:DISPLAY:FORMAT | HEXADECIMAL |
| BUS:B2:DISPLAY:TYPE | BUS |
| BUS:B2:I2C:ADDRESS:RWINCLUDE | 0 |
| BUS:B2:I2C:CLOCK:SOURCE | CH1 |
| BUS:B2:I2C:DATA:SOURCE | CH2 |
| BUS:B2:LABEL | "Parallel" |
| BUS:B2:LIN:BITRATE | 19200 |
| BUS:B2:LIN:IDFORMAT | NOPARITY |
| BUS:B2:LIN:POLARITY | NORMAL |
| BUS:B2:LIN:SAMPLEPOINT | 50 |
| BUS:B2:LIN:SOURCE | CH1 |
| BUS:B2:LIN:STANDARD | V2X |
| BUS:B2:PARALLEL:BIT0:SOURCE | D0 |
| BUS:B2:PARALLEL:BIT10:SOURCE | D10 |
| BUS:B2:PARALLEL:BIT11:SOURCE | D11 |
| BUS:B2:PARALLEL:BIT12:SOURCE | D12 |
| BUS:B2:PARALLEL:BIT13:SOURCE | D13 |
| BUS:B2:PARALLEL:BIT14:SOURCE | D14 |
| BUS:B2:PARALLEL:BIT15:SOURCE | D15 |
| BUS:B2:PARALLEL:BIT16:SOURCE | CH1 |
| BUS:B2:PARALLEL:BIT17:SOURCE | CH2 |
| BUS:B2:PARALLEL:BIT18:SOURCE | CH3 |
| BUS:B2:PARALLEL:BIT19:SOURCE | CH4 |
| BUS:B2:PARALLEL:BIT1:SOURCE | D1 |
| BUS:B2:PARALLEL:BIT2:SOURCE | D2 |
| BUS:B2:PARALLEL:BIT3:SOURCE | D3 |
| BUS:B2:PARALLEL:BIT4:SOURCE | D4 |
| BUS:B2:PARALLEL:BIT5:SOURCE | D5 |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------------------------|-------------|
| BUS:B2:PARALLEL:BIT6:SOURCE | D6 |
| BUS:B2:PARALLEL:BIT7:SOURCE | D7 |
| BUS:B2:PARALLEL:BIT8:SOURCE | D8 |
| BUS:B2:PARALLEL:BIT9:SOURCE | D9 |
| BUS:B2:PARALLEL:CLOCK:EDGE | RISING |
| BUS:B2:PARALLEL:CLOCK:ISCLOCKED | NO |
| BUS:B2:PARALLEL:CLOCK:SOURCE | CH1 |
| BUS:B2:PARALLEL:WIDTH | 16 |
| BUS:B2:POSITION | 0.0E+0 |
| BUS:B2:RS232C:BITRATE | 9600 |
| BUS:B2:RS232C:DATABITS | 8 |
| BUS:B2:RS232C:DELIMITER | LF |
| BUS:B2:RS232C:DISPLAYMODE | FRAME |
| BUS:B2:RS232C:PARITY | NONE |
| BUS:B2:RS232C:POLARITY | NORMAL |
| BUS:B2:RS232C:RX:SOURCE | OFF |
| BUS:B2:RS232C:TX:SOURCE | CH1 |
| BUS:B2:SPI:BITORDER | MSB |
| BUS:B2:SPI:CLOCK:POLARITY | RISE |
| BUS:B2:SPI:CLOCK:SOURCE | CH1 |
| BUS:B2:SPI:DATA:IN:POLARITY | HIGH |
| BUS:B2:SPI:DATA:IN:SOURCE | OFF |
| BUS:B2:SPI:DATA:OUT:POLARITY | HIGH |
| BUS:B2:SPI:DATA:OUT:SOURCE | CH3 |
| BUS:B2:SPI:DATA:SIZE | 8 |
| BUS:B2:SPI:FRAMING | SS |
| BUS:B2:SPI:IDLETIME | 5.0000E-6 |
| BUS:B2:SPI:SELECT:POLARITY | LOW |
| BUS:B2:SPI:SELECT:SOURCE | CH2 |
| BUS:B2:STATE | 0 |
| BUS:B2:TYPE | PARALLEL |
| BUS:LOWERTHRESHOLD:CH1 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:CH2 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:CH3 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:CH4 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:MATH | 0.0E+0 |
| BUS:LOWERTHRESHOLD:REF1 | 0.0E+0 |

Table C-1: Default Values (cont.)

| Item | Description |
|-------------------------|-------------|
| BUS:LOWERTHRESHOLD:REF2 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:REF3 | 0.0E+0 |
| BUS:LOWERTHRESHOLD:REF4 | 0.0E+0 |
| BUS:UPPERTHRESHOLD:CH1 | 800.0000E-3 |
| BUS:UPPERTHRESHOLD:CH2 | 800.0000E-3 |
| BUS:UPPERTHRESHOLD:CH3 | 800.0000E-3 |
| BUS:UPPERTHRESHOLD:CH4 | 800.0000E-3 |
| BUS:UPPERTHRESHOLD:MATH | 0.0E+0 |
| BUS:UPPERTHRESHOLD:REF1 | 0.0E+0 |
| BUS:UPPERTHRESHOLD:REF2 | 0.0E+0 |
| BUS:UPPERTHRESHOLD:REF3 | 0.0E+0 |
| BUS:UPPERTHRESHOLD:REF4 | 0.0E+0 |
| CH1:AMPSVIAVOLTS:ENABLE | 0 |
| CH1:AMPSVIAVOLTS:FACTOR | 10.0000 |
| CH1:BANDWIDTH | 1.0000E+9 |
| CH1:COUPLING | DC |
| CH1:DESKEW | 0.0E+0 |
| CH1:INVERT | 0 |
| CH1:LABEL | "" |
| CH1:OFFSET | 0.0E+0 |
| CH1:POSITION | 0.0E+0 |
| CH1:SCALE | 100.0000E-3 |
| CH1:TERMINATION | 1.0000E+6 |
| CH1:YUNITS | "V" |
| CH2:AMPSVIAVOLTS:ENABLE | 0 |
| CH2:AMPSVIAVOLTS:FACTOR | 10.0000 |
| CH2:BANDWIDTH | 1.0000E+9 |
| CH2:COUPLING | DC |
| CH2:DESKEW | 0.0E+0 |
| CH2:INVERT | 0 |
| CH2:LABEL | "" |
| CH2:OFFSET | 0.0E+0 |
| CH2:POSITION | 0.0E+0 |
| CH2:SCALE | 100.0000E-3 |
| CH2:TERMINATION | 1.0000E+6 |
| CH2:YUNITS | "V" |
| CH3:AMPSVIAVOLTS:ENABLE | 0 |

Table C-1: Default Values (cont.)

| Item | Description |
|-----------------------------------|-------------|
| CH3:AMPSVIAVOLTS:FACTOR | 10.0000 |
| CH3:BANDWIDTH | 1.0000E+9 |
| CH3:COUPLING | DC |
| CH3:DESKEW | 0.0E+0 |
| CH3:INVERT | 0 |
| CH3:LABEL | "" |
| CH3:OFFSET | 0.0E+0 |
| CH3:POSITION | 0.0E+0 |
| CH3:SCALE | 100.0000E-3 |
| CH3:TERMINATION | 1.0000E+6 |
| CH3:YUNITS | "V" |
| CH4:AMPSVIAVOLTS:ENABLE | 0 |
| CH4:AMPSVIAVOLTS:FACTOR | 10.0000 |
| CH4:BANDWIDTH | 1.0000E+9 |
| CH4:COUPLING | DC |
| CH4:DESKEW | 0.0E+0 |
| CH4:INVERT | 0 |
| CH4:LABEL | "" |
| CH4:OFFSET | 0.0E+0 |
| CH4:POSITION | 0.0E+0 |
| CH4:SCALE | 100.0000E-3 |
| CH4:TERMINATION | 1.0000E+6 |
| CH4:YUNITS | "V" |
| CURSOR:FUNCTION | OFF |
| CURSOR:HBARS:POSITION1 | 0.0E+0 |
| CURSOR:HBARS:POSITION2 | 0.0E+0 |
| CURSOR:HBARS:UNITS | BASE |
| CURSOR:MODE | INDEPENDENT |
| CURSOR:SOURCE | AUTO |
| CURSOR:VBARS:POSITION1 | -12.00E-6 |
| CURSOR:VBARS:POSITION2 | 12.00E-6 |
| CURSOR:VBARS:UNITS | SECONDS |
| CURSOR:XY:READOUT | RECTANGULAR |
| CURSOR:XY:RECTANGULAR:X:POSITION1 | 0.0E+0 |
| CURSOR:XY:RECTANGULAR:X:POSITION2 | 0.0E+0 |
| CURSOR:XY:RECTANGULAR:Y:POSITION1 | 0.0E+0 |
| CURSOR:XY:RECTANGULAR:Y:POSITION2 | 0.0E+0 |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------|--------------------|
| D0:LABEL | "" |
| D0:POSITION | 80.0000E-3 |
| D0:THRESHOLD | 1.4000 |
| D10:LABEL | "" |
| D10:POSITION | 80.0000E-3 |
| D10:THRESHOLD | 1.4000 |
| D11:LABEL | "" |
| D11:POSITION | 80.0000E-3 |
| D11:THRESHOLD | 1.4000 |
| D12:LABEL | "" |
| D12:POSITION | 80.0000E-3 |
| D12:THRESHOLD | 1.4000 |
| D13:LABEL | "" |
| D13:POSITION | 80.0000E-3 |
| D13:THRESHOLD | 1.4000 |
| D14:LABEL | "" |
| D14:POSITION | 80.0000E-3 |
| D14:THRESHOLD | 1.4000 |
| D15:LABEL | "" |
| D15:POSITION | 80.0000E-3 |
| D15:THRESHOLD | 1.4000 |
| D1:LABEL | "" |
| D1:POSITION | 80.0000E-3 |
| D1:THRESHOLD | 1.4000 |
| D2:LABEL | "" |
| D2:POSITION | 80.0000E-3 |
| D2:THRESHOLD | 1.4000 |
| D3:LABEL | "" |
| D3:POSITION | 80.0000E-3 |
| D3:THRESHOLD | 1.4000 |
| D4:LABEL | "" |
| D4:POSITION | 80.0000E-3 |
| D4:THRESHOLD | 1.4000 |
| D5:LABEL | "" |
| D5:POSITION | 80.0000E-3 |
| D5:THRESHOLD | 1.4000 |
| D6:LABEL | "" |

Table C-1: Default Values (cont.)

| Item | Description |
|--|--------------|
| D6:POSITION | 80.0000E-3 |
| D6:THRESHOLD | 1.4000 |
| D7:LABEL | "" |
| D7:POSITION | 80.0000E-3 |
| D7:THRESHOLD | 1.4000 |
| D8:LABEL | "" |
| D8:POSITION | 80.0000E-3 |
| D8:THRESHOLD | 1.4000 |
| D9:LABEL | "" |
| D9:POSITION | 80.0000E-3 |
| D9:THRESHOLD | 1.4000 |
| DATA:DESTINATION | REF1 |
| DATA:ENCDG | RIBINARY |
| DATA:SOURCE | CH1 |
| DATA:START | 1 |
| DATA:STOP | 10000 |
| DESKEW:DISPLAY | 1 |
| DISPLAY:CLOCK | 1 |
| DISPLAY:DIGITAL:ACTIVITY | 0 |
| DISPLAY:DIGITAL:HEIGHT | MEDIUM |
| DISPLAY:GRATICULE | FULL |
| DISPLAY:INTENSITY:BACKLIGHT | HIGH |
| DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE | 1 |
| DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME | 60 |
| DISPLAY:INTENSITY:GRATICULE | 75 |
| DISPLAY:INTENSITY:WAVEFORM | 35 |
| DISPLAY:PERSISTENCE | 0.0E+0 |
| DISPLAY:STYLE:DOTSONLY | 0 |
| DISPLAY:TRIGFREQUENCY | 0 |
| DISPLAY:XY | OFF |
| DISPLAY:XY:WITHYT | 0 |
| DVM:AUTORANGE | 0 |
| DVM:DISPLAYSTYLE | FULL |
| DVM:MODE | OFF |
| DVM:SOURCE | CH1 |
| EMAIL:SETUP:FROMADDRESS | "TestString" |
| EMAIL:SETUP:HOSTALIASNAME | "TestString" |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------------------|---------------------------------|
| EMAIL:SETUP:SMTPLGIN | "TestString" |
| EMAIL:SETUP:SMTPPORT | 25 |
| EMAIL:SETUP:SMTPSERVER | "TestString" |
| FASTACQ:STATE | 0 |
| HEADER | 0 |
| HISTOGRAM:BOXPCNT | 20.0000,20.0000,80.0000,80.0000 |
| HISTOGRAM:DISPLAY | LINEAR |
| HISTOGRAM:MODE | OFF |
| HISTOGRAM:SOURCE | CH1 |
| HORIZONTAL:DELAY:MODE | 1 |
| HORIZONTAL:DELAY:TIME | 0.0E+0 |
| HORIZONTAL:POSITION | 50.0000 |
| HORIZONTAL:RECORDLENGTH | 10000 |
| | 4.0000E-6 |
| LOCK | NONE |
| MARKER:M1:FREQUENCY:ABSOLUTE | 0.0E+0 |
| MARKER:M2:FREQUENCY:ABSOLUTE | 0.0E+0 |
| MARKER:MANUAL | 0 |
| MARKER:PEAK:EXCURSION | 30.0000 |
| MARKER:PEAK:MAXIMUM | 5 |
| MARKER:PEAK:STATE | 1 |
| MARKER:PEAK:THRESHOLD | -50.0000 |
| MARKER:TYPE | ABSOLUTE |
| MATH:AUTOSCALE | 1 |
| MATH:DEFINE | "CH1+CH2" |
| MATH:HORIZONTAL:POSITION | 50.0000 |
| MATH: | 4.0000E-6 |
| MATH:HORIZONTAL:UNITS | "s" |
| MATH:LABEL | "" |
| MATH:SPECTRAL:MAG | DB |
| MATH:SPECTRAL:WINDOW | HANNING |
| MATH:TYPE | DUAL |
| MATH:VERTICAL:POSITION | 0.0E+0 |
| MATH:VERTICAL:SCALE | 100.0000E-3 |
| MATH:VERTICAL:UNITS | "V" |
| MATHVAR:VAR1 | 0.0E+0 |
| MATHVAR:VAR2 | 0.0E+0 |

Table C-1: Default Values (cont.)

| Item | Description |
|-----------------------------------|-------------|
| MEASUREMENT:GATING | SCREEN |
| MEASUREMENT:IMMED:DELAY:DIRECTION | FORWARDS |
| MEASUREMENT:IMMED:DELAY:EDGE1 | RISE |
| MEASUREMENT:IMMED:DELAY:EDGE2 | RISE |
| MEASUREMENT:IMMED:SOURCE1 | CH1 |
| MEASUREMENT:IMMED:SOURCE2 | CH2 |
| MEASUREMENT:IMMED:TYPE | PERIOD |
| MEASUREMENT:INDICATORS:STATE | OFF |
| MEASUREMENT:MEAS1:DELAY:DIRECTION | FORWARDS |
| MEASUREMENT:MEAS1:DELAY:EDGE1 | RISE |
| MEASUREMENT:MEAS1:DELAY:EDGE2 | RISE |
| MEASUREMENT:MEAS1:SOURCE1 | CH1 |
| MEASUREMENT:MEAS1:SOURCE2 | CH2 |
| MEASUREMENT:MEAS1:STATE | 0 |
| MEASUREMENT:MEAS1:TYPE | PERIOD |
| MEASUREMENT:MEAS2:DELAY:DIRECTION | FORWARDS |
| MEASUREMENT:MEAS2:DELAY:EDGE1 | RISE |
| MEASUREMENT:MEAS2:DELAY:EDGE2 | RISE |
| MEASUREMENT:MEAS2:SOURCE1 | CH1 |
| MEASUREMENT:MEAS2:SOURCE2 | CH2 |
| MEASUREMENT:MEAS2:STATE | 0 |
| MEASUREMENT:MEAS2:TYPE | PERIOD |
| MEASUREMENT:MEAS3:DELAY:DIRECTION | FORWARDS |
| MEASUREMENT:MEAS3:DELAY:EDGE1 | RISE |
| MEASUREMENT:MEAS3:DELAY:EDGE2 | RISE |
| MEASUREMENT:MEAS3:SOURCE1 | CH1 |
| MEASUREMENT:MEAS3:SOURCE2 | CH2 |
| MEASUREMENT:MEAS3:STATE | 0 |
| MEASUREMENT:MEAS3:TYPE | PERIOD |
| MEASUREMENT:MEAS4:DELAY:DIRECTION | FORWARDS |
| MEASUREMENT:MEAS4:DELAY:EDGE1 | RISE |
| MEASUREMENT:MEAS4:DELAY:EDGE2 | RISE |
| MEASUREMENT:MEAS4:SOURCE1 | CH1 |
| MEASUREMENT:MEAS4:SOURCE2 | CH2 |
| MEASUREMENT:MEAS4:STATE | 0 |
| MEASUREMENT:MEAS4:TYPE | PERIOD |
| MEASUREMENT:METHOD | AUTO |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------------------------|---------------|
| MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH | 0.0E+0 |
| MEASUREMENT:REFLEVEL:ABSOLUTE:LOW | 0.0E+0 |
| MEASUREMENT:REFLEVEL:ABSOLUTE:MID1 | 0.0E+0 |
| MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 | 0.0E+0 |
| MEASUREMENT:REFLEVEL:METHOD | PERCENT |
| MEASUREMENT:REFLEVEL:PERCENT:HIGH | 90.0000 |
| MEASUREMENT:REFLEVEL:PERCENT:LOW | 10.0000 |
| MEASUREMENT:REFLEVEL:PERCENT:MID1 | 50.0000 |
| MEASUREMENT:REFLEVEL:PERCENT:MID2 | 50.0000 |
| MEASUREMENT:STATISTICS:MODE | ALL |
| MEASUREMENT:STATISTICS:WEIGHTING | 32 |
| MESSAGE:BOX | 167,67,167,83 |
| MESSAGE:SHOW | "" |
| MESSAGE:STATE | 0 |
| PICTBRIDGE:DATEPRINT | DEFLT |
| PICTBRIDGE:IDPRINT | OFF |
| PICTBRIDGE:IMAGESIZE | DEFLT |
| PICTBRIDGE:PAPERSIZE | DEFLT |
| PICTBRIDGE:PAPERTYPE | DEFLT |
| PICTBRIDGE:PRINTQUAL | DEFLT |
| REF1:HORIZONTAL:DELAY:TIME | 16.0000E-6 |
| REF1: | 4.0000E-6 |
| REF1:VERTICAL:POSITION | 0.0E+0 |
| REF1:VERTICAL:SCALE | 100.0000E-3 |
| REF2:HORIZONTAL:DELAY:TIME | 16.0000E-6 |
| REF2: | 4.0000E-6 |
| REF2:VERTICAL:POSITION | 0.0E+0 |
| REF2:VERTICAL:SCALE | 100.0000E-3 |
| REF3:HORIZONTAL:DELAY:TIME | 16.0000E-6 |
| REF3: | 4.0000E-6 |
| REF3:VERTICAL:POSITION | 0.0E+0 |
| REF3:VERTICAL:SCALE | 100.0000E-3 |
| REF4:HORIZONTAL:DELAY:TIME | 16.0000E-6 |
| REF4: | 4.0000E-6 |
| REF4:VERTICAL:POSITION | 0.0E+0 |
| REF4:VERTICAL:SCALE | 100.0000E-3 |
| ROSC:SOURCE | INTERNAL |

Table C-1: Default Values (cont.)

| Item | Description |
|--|------------------------------|
| SAVE:ASSIGN:TYPE | WAVEFORM |
| SAVE:IMAGE:FILEFORMAT | PNG |
| SAVE:IMAGE:INKSAVER | 1 |
| SAVE:IMAGE:LAYOUT | PORTRAIT |
| SAVE:WAVEFORM | WFM |
| SAVE:WAVEFORM:GATING | NONE |
| SEARCH:SEARCH1:STATE | 0 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:CONDITION | SOF |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET | 0 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:WORD | EITHER |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:CONDITION | SOF |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION | NOCARE |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:OFFSET | 0 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:VALUE | "XXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:BRISBIT | X |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:ESIBIT | X |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FRAMETYPE | DATA |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE | STANDARD |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE | "XXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE | ADDR7 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE | USER |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE | "XXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:CONDITION | START |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION | NOCARE |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:VALUE | "XXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:CONDITION | SYNCFIELD |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE | "XXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:VALUE | "XXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:ERRTYPE | SYNC |

Table C-1: Default Values (cont.)

| Item | Description |
|--|------------------------------|
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE | "XXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:PARALLEL:VALUE | "XXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:CONDITION | TXSTART |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:CONDITION | SS |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MISO:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MOSI:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:CONDITION | SOF |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:HIVALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:OFFSET | 0 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:VALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:WORD | EITHER |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:CONDITION | SOF |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION | NOCARE |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:OFFSET | 0 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:BRSBIT | X |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:ESIBIT | X |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FRAMETYPE | DATA |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE | STANDARD |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE | "XXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE | ADDR7 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE | USER |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE | "XXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:CONDITION | START |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION | NOCARE |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:CONDITION | SYNCFIELD |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE | "XXXXXXXX" |

Table C-1: Default Values (cont.)

| Item | Description |
|--|----------------------|
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER | EQUAL |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:ERRTYPE | SYNC |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE | "XXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:PARALLEL:VALUE | "XXXXXXXXXXXXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:CONDITION | TXSTART |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:CONDITION | SS |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MISO:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MOSI:VALUE | "XXXXXXXX" |
| SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:SIZE | 1 |
| SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE | B1 |
| SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE | RISE |
| SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE | CH1 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION | AND |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH2 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH3 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH4 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE | RISE |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE | NONE |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D0 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D1 | X |

Table C-1: Default Values (cont.)

| Item | Description |
|---|-------------|
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D10 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D11 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D12 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D13 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D14 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D15 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D2 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D3 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D4 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D5 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D6 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D7 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D8 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D9 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF2 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF3 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF4 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D0 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D1 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D10 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D11 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D12 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D13 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D14 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D15 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D2 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D3 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D4 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D5 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D6 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D7 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D8 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D9 | X |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN | TRUE |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT | 8.0000E-9 |

Table C-1: Default Values (cont.)

| Item | Description |
|---|-------------|
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:MATH | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT | 12.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY | POSITIVE |
| SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN | LESSTHAN |
| SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WIDTH | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY | POSITIVE |
| SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN | OCCURS |
| SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE | RISE |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE | CH1 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE | NONE |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD | 9.91E+37 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH3 | 0.0E+0 |

Table C-1: Default Values (cont.)

| Item | Description |
|---|--------------------------------|
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:MATH | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF1 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF2 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF3 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF4 | 0.0E+0 |
| SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY | STAYSHIGH |
| SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME | 8.0000E-9 |
| SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY | POSITIVE |
| SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN | SLOWER |
| SEARCH:SEARCH1:TRIGGER:A:TYPE | EDGE |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH1 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH2 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH3 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH4 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:MATH | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF1 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF2 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF3 | 800.0000E-3 |
| SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF4 | 800.0000E-3 |
| SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:OFFSet | 0 |
| SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:BRSEBIT | X |
| SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT | X |
| SELECT:CH1 | 1 |
| SELECT:CH2 | 0 |
| SELECT:CH3 | 0 |
| SELECT:CH4 | 0 |
| TRIGGER:A:BUS:B1:AUDIO:CONDITION | SOF |
| TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXXXX" |
| TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET | 0 |
| TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER | EQUAL |
| TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE | "XXXXXXXXXXXXXXXXXXXXXXXXXXXX" |
| TRIGGER:A:BUS:B1:AUDIO:DATA:WORD | EITHER |
| TRIGGER:A:BUS:B1:CAN:CONDITION | SOF |
| TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION | NO CARE |
| TRIGGER:A:BUS:B1:CAN:DATA:OFFSET | 0 |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------------------------------|------------------------|
| TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER | EQUAL |
| TRIGGER:A:BUS:B1:CAN:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:CAN:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B1:CAN:FD:BRSBIT | X |
| TRIGGER:A:BUS:B1:CAN:FD:ESIBIT | X |
| TRIGGER:A:BUS:B1:CAN:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:CAN:FRAMETYPE | DATA |
| TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE | STANDARD |
| TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE | "XXXXXXXXXXXX" |
| TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE | ADDR7 |
| TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE | USER |
| TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE | "XXXXXXX" |
| TRIGGER:A:BUS:B1:I2C:CONDITION | START |
| TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION | NOCARE |
| TRIGGER:A:BUS:B1:I2C:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:I2C:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B1:I2C:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:LIN:CONDITION | SYNCFIELD |
| TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER | EQUAL |
| TRIGGER:A:BUS:B1:LIN:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:LIN:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:LIN:ERRTYPE | SYNC |
| TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE | "XXXXXX" |
| TRIGGER:A:BUS:B1:PARALLEL:VALUE | "XXXXXXXXXXXXXXXXXXXX" |
| TRIGGER:A:BUS:B1:RS232C:CONDITION | TXSTART |
| TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:SPI:CONDITION | SS |
| TRIGGER:A:BUS:B1:SPI:DATA:IN:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:SPI:DATA:OUT:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B1:SPI:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B1:SPI:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B2:CAN:CONDITION | SOF |
| TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION | NOCARE |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------------------------------|------------------------|
| TRIGGER:A:BUS:B2:CAN:DATA:OFFSET | 0 |
| TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER | EQUAL |
| TRIGGER:A:BUS:B2:CAN:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:CAN:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B2:CAN:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:CAN:FD:BRSBUT | X |
| TRIGGER:A:BUS:B2:CAN:FD:ESIBIT | X |
| TRIGGER:A:BUS:B2:CAN:FRAMETYPE | DATA |
| TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE | STANDARD |
| TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE | "XXXXXXXXXXXX" |
| TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE | ADDR7 |
| TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE | USER |
| TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE | "XXXXXXX" |
| TRIGGER:A:BUS:B2:I2C:CONDITION | START |
| TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION | NO CARE |
| TRIGGER:A:BUS:B2:I2C:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:I2C:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B2:I2C:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:LIN:CONDITION | SYNCFIELD |
| TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER | EQUAL |
| TRIGGER:A:BUS:B2:LIN:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:LIN:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:LIN:ERRTYPE | SYNC |
| TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE | "XXXXXX" |
| TRIGGER:A:BUS:B2:PARALLEL:VALUE | "XXXXXXXXXXXXXXXXXXXX" |
| TRIGGER:A:BUS:B2:RS232C:CONDITION | TXSTART |
| TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:SPI:CONDITION | SS |
| TRIGGER:A:BUS:B2:SPI:DATA:IN:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:SPI:DATA:OUT:VALUE | "XXXXXXXX" |
| TRIGGER:A:BUS:B2:SPI:DATA:SIZE | 1 |
| TRIGGER:A:BUS:B2:SPI:DATA:START | 0.0E+0 |
| TRIGGER:A:BUS:B<x>:CAN:DATA:OFFSet | 0 |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------------------------|-------------|
| TRIGGER:A:BUS:B<x>:CAN:FD:BRSBIT | X |
| TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT | X |
| TRIGGER:A:BUS:SOURCE | B1 |
| TRIGGER:A:EDGE:COUPLING | DC |
| TRIGGER:A:EDGE:SLOPE | RISE |
| TRIGGER:A:EDGE:SOURCE | CH1 |
| TRIGGER:A:HOLDOFF:TIME | 20.0000E-9 |
| TRIGGER:A:LEVEL | 0.0E+0 |
| TRIGGER:A:LEVEL:CH1 | 0.0E+0 |
| TRIGGER:A:LEVEL:CH2 | 0.0E+0 |
| TRIGGER:A:LEVEL:CH3 | 0.0E+0 |
| TRIGGER:A:LEVEL:CH4 | 0.0E+0 |
| TRIGGER:A:LEVEL:D0 | 1.4000 |
| TRIGGER:A:LEVEL:D1 | 1.4000 |
| TRIGGER:A:LEVEL:D10 | 1.4000 |
| TRIGGER:A:LEVEL:D11 | 1.4000 |
| TRIGGER:A:LEVEL:D12 | 1.4000 |
| TRIGGER:A:LEVEL:D13 | 1.4000 |
| TRIGGER:A:LEVEL:D14 | 1.4000 |
| TRIGGER:A:LEVEL:D15 | 1.4000 |
| TRIGGER:A:LEVEL:D2 | 1.4000 |
| TRIGGER:A:LEVEL:D3 | 1.4000 |
| TRIGGER:A:LEVEL:D4 | 1.4000 |
| TRIGGER:A:LEVEL:D5 | 1.4000 |
| TRIGGER:A:LEVEL:D6 | 1.4000 |
| TRIGGER:A:LEVEL:D7 | 1.4000 |
| TRIGGER:A:LEVEL:D8 | 1.4000 |
| TRIGGER:A:LEVEL:D9 | 1.4000 |
| TRIGGER:A:LOGIC:CLASS | LOGIC |
| TRIGGER:A:LOGIC:FUNCTION | AND |
| TRIGGER:A:LOGIC:INPUT:CH1 | X |
| TRIGGER:A:LOGIC:INPUT:CH2 | X |
| TRIGGER:A:LOGIC:INPUT:CH3 | X |
| TRIGGER:A:LOGIC:INPUT:CH4 | X |
| TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE | RISE |
| TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE | NONE |
| TRIGGER:A:LOGIC:INPUT:D0 | X |

Table C-1: Default Values (cont.)

| Item | Description |
|--|-------------|
| TRIGGER:A:LOGIC:INPUT:D1 | X |
| TRIGGER:A:LOGIC:INPUT:D10 | X |
| TRIGGER:A:LOGIC:INPUT:D11 | X |
| TRIGGER:A:LOGIC:INPUT:D12 | X |
| TRIGGER:A:LOGIC:INPUT:D13 | X |
| TRIGGER:A:LOGIC:INPUT:D14 | X |
| TRIGGER:A:LOGIC:INPUT:D15 | X |
| TRIGGER:A:LOGIC:INPUT:D2 | X |
| TRIGGER:A:LOGIC:INPUT:D3 | X |
| TRIGGER:A:LOGIC:INPUT:D4 | X |
| TRIGGER:A:LOGIC:INPUT:D5 | X |
| TRIGGER:A:LOGIC:INPUT:D6 | X |
| TRIGGER:A:LOGIC:INPUT:D7 | X |
| TRIGGER:A:LOGIC:INPUT:D8 | X |
| TRIGGER:A:LOGIC:INPUT:D9 | X |
| TRIGGER:A:LOGIC:PATTERN:DELTATIME | 8.0000E-9 |
| TRIGGER:A:LOGIC:PATTERN:WHEN | TRUE |
| TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT | 8.0000E-9 |
| TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT | 8.0000E-9 |
| TRIGGER:A:LOGIC:THRESHOLD:CH1 | 0.0E+0 |
| TRIGGER:A:LOGIC:THRESHOLD:CH2 | 0.0E+0 |
| TRIGGER:A:LOGIC:THRESHOLD:CH3 | 0.0E+0 |
| TRIGGER:A:LOGIC:THRESHOLD:CH4 | 0.0E+0 |
| TRIGGER:A:LOGIC:THRESHOLD:D0 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D1 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D10 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D11 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D12 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D13 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D14 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D15 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D2 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D3 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D4 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D5 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D6 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D7 | 1.4000 |

Table C-1: Default Values (cont.)

| Item | Description |
|-----------------------------------|-------------|
| TRIGGER:A:LOGIC:THRESHOLD:D8 | 1.4000 |
| TRIGGER:A:LOGIC:THRESHOLD:D9 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:CH1 | 0.0E+0 |
| TRIGGER:A:LOWERTHRESHOLD:CH2 | 0.0E+0 |
| TRIGGER:A:LOWERTHRESHOLD:CH3 | 0.0E+0 |
| TRIGGER:A:LOWERTHRESHOLD:CH4 | 0.0E+0 |
| TRIGGER:A:LOWERTHRESHOLD:D0 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D1 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D10 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D11 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D12 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D13 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D14 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D15 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D2 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D3 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D4 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D5 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D6 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D7 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D8 | 1.4000 |
| TRIGGER:A:LOWERTHRESHOLD:D9 | 1.4000 |
| TRIGGER:A:MODE | AUTO |
| TRIGGER:A:PULSE:CLASS | WIDTH |
| TRIGGER:A:PULSEWIDTH:HIGHLIMIT | 12.0000E-9 |
| TRIGGER:A:PULSEWIDTH:LOWLIMIT | 8.0000E-9 |
| TRIGGER:A:PULSEWIDTH:POLARITY | POSITIVE |
| TRIGGER:A:PULSEWIDTH:WHEN | LESSTHAN |
| TRIGGER:A:PULSEWIDTH:WIDTH | 8.0000E-9 |
| TRIGGER:A:RUNT:POLARITY | POSITIVE |
| TRIGGER:A:RUNT:WHEN | OCCURS |
| TRIGGER:A:RUNT:WIDTH | 8.0000E-9 |
| TRIGGER:A:SETHOLD:CLOCK:EDGE | RISE |
| TRIGGER:A:SETHOLD:CLOCK:SOURCE | CH1 |
| TRIGGER:A:SETHOLD:CLOCK:THRESHOLD | 0.0E+0 |
| TRIGGER:A:SETHOLD:DATA:SOURCE | NONE |
| TRIGGER:A:SETHOLD:DATA:THRESHOLD | 9.91E+37 |

Table C-1: Default Values (cont.)

| Item | Description |
|---------------------------------|-------------|
| TRIGGER:A:SETHOLD:HOLDTIME | 8.0000E-9 |
| TRIGGER:A:SETHOLD:SETTIME | 8.0000E-9 |
| TRIGGER:A:SETHOLD:THRESHOLD:CH1 | 0.0E+0 |
| TRIGGER:A:SETHOLD:THRESHOLD:CH2 | 0.0E+0 |
| TRIGGER:A:SETHOLD:THRESHOLD:CH3 | 0.0E+0 |
| TRIGGER:A:SETHOLD:THRESHOLD:CH4 | 0.0E+0 |
| TRIGGER:A:SETHOLD:THRESHOLD:D0 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D1 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D10 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D11 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D12 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D13 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D14 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D15 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D2 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D3 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D4 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D5 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D6 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D7 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D8 | 1.4000 |
| TRIGGER:A:SETHOLD:THRESHOLD:D9 | 1.4000 |
| TRIGGER:A:TIMEOUT:POLARITY | STAYSHIGH |
| TRIGGER:A:TIMEOUT:TIME | 8.0000E-9 |
| TRIGGER:A:TRANSITION:DELTATIME | 8.0000E-9 |
| TRIGGER:A:TRANSITION:POLARITY | POSITIVE |
| TRIGGER:A:TRANSITION:WHEN | SLOWER |
| TRIGGER:A:TYPE | EDGE |
| TRIGGER:A:UPPERTHRESHOLD:CH1 | 800.0000E-3 |
| TRIGGER:A:UPPERTHRESHOLD:CH2 | 800.0000E-3 |
| TRIGGER:A:UPPERTHRESHOLD:CH3 | 800.0000E-3 |
| TRIGGER:A:UPPERTHRESHOLD:CH4 | 800.0000E-3 |
| VERBOSE | 1 |
| VIDPIC:AUTOCONTRAST | 1 |
| VIDPIC:AUTOCONTRAST:UPDATERATE | 20 |
| ZOOM:MODE | 0 |
| ZOOM:ZOOM1:POSITION | 50.0000 |

Table C-1: Default Values (cont.)

| Item | Description |
|------------------|--------------------|
| ZOOM:ZOOM1:SCALE | 4.0000E-6 |
| ZOOM:ZOOM1:STATE | 0 |

Appendix D: Waveform Transfer (WFMOutpre? and CURVe? Query) Examples

Here are several examples of command sequences using the WFMOutpre? and CURVe? queries to transfer data from the instrument to a PC, with different data sources, including Analog, Digital waveforms. Each command sequence example is followed by an explanation of the returned WFMOutpre? results. The WFMOutpre? values act as settings that apply to the CURVe? query data being transferred.

NOTE. In order to guarantee that the waveform data returned from CURVe? queries of multiple waveforms are correlated to the same acquisition, you should use single sequence acquisition mode to acquire the waveform data from a single acquisition. Single sequence acquisition mode is enabled using SEQUENCE.

Example 1: Analog Waveform (Channels 1–4)

Goal: Transfer 10,000 points of analog channel waveform data from the instrument to a PC.

| Command | Comment |
|-------------------------|--|
| :DATA:SOURce CH1 | |
| :DATA:START 1 | |
| :DATA:STOP 10000 | |
| :WFMOutpre:ENCdg BINARY | |
| :WFMOutpre:BYT_Nr 1 | |
| :HEADer 1 | |
| :WFMOutpre? | Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):
:WFMOUTPRE:BYT_NR 1;BIT_NR 8;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode";NR_PT 10000;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 4.0000E-9;XZERO 139.9999E-12;PT_OFF 5000;YUNIT "v";YMULT 4.0000E-3;YOFF 0.0E+0;YZERO 0.0E+0 |
| :CURVe? | Returns 10,000 data points:
:CURVe #510000<10,000 binary bytes of waveform data.> |

NOTE. You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.

| WFMOupre? Query results | Explanation |
|--|--|
| BYT_NR 1 | This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOupre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the command.) |
| BIT_NR 8 | This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOupre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly. |
| ENCDG BINARY | This value specifies the encoding of the waveform data. To change this value (the other possibility is ASCii), use the WFMOupre:ENCDg command. (This value can also be set using the DATA:ENCDg command, which provides the ability to set the WFMOupre:ENCDg , WFMOupre:BN_Fmt , and WFMOupre:BYT_Or values using a single command.) |
| BN_FMT RI | This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOupre:BN_Fmt command. Note: this field is not applicable for ASCii encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first). To change this value to LSB, use the WFMOupre:BYT_Or command. Note: this field is not applicable for ASCii encoding. |
| WFID "Ch1, DC coupling, 100.0mv/div, 4.000us/div, 10000 points, Sample mode" | This quoted string represents information about the source waveform that would be returned by a WFMOupre:WFid? query. |
| NR_PT 10000 | This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOupre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOupre:PT_Fmt? query.) |
| PT_ORDER LINEAR | This value is LINEAR or ROW. |
| XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOupre:XUNit? query.) |
| XINCR 4.0000E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOupre:XINcr? query.) |
| XZERO 139.9999E-12 | This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOupre:XUNit . This is query only. (If you would like to determine only this value, use the WFMOupre:XZEro? query.) |
| PT_OFF 5000 | This value indicates the data point immediately following the trigger point relative to DATA:START in the waveform record. This is a query only. (If you would like to determine only this value, use the WFMOupre:PT_Off? query.) |

| WFMOUtpre? Query results | Explanation |
|--------------------------|--|
| YUNIT "V" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, volts. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YUNIT? query.) |
| YMULT 4.0000E-3 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the <code>WFMOUtpre:YUNIT</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YMULT? query.) |
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YOFF? query.) |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the <code>WFMOUtpre:YUNIT</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YZERO? query.) |

Example 2: Digital Waveform

Goal: Transfer 25 points of digital channel waveform data from the instrument to a PC.

| Command | Comment |
|-------------------------------------|---|
| <code>:DATA:SOURce CH1_D5</code> | |
| <code>:DATA:START 1</code> | |
| <code>:DATA:STOP 25</code> | |
| <code>:WFMOUtpre:ENCdg ASCii</code> | |
| <code>:WFMOUtpre:BYT_Nr 1</code> | |
| <code>:HEADer 1</code> | |
| <code>:VERBose 1</code> | |
| <code>:WFMOUtpre?</code> | Returns the following values. Each value represents the current settings that a <code>CURVe?</code> query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):
<code>:WFMOUtpre:BYT_NR 1;BIT_NR 8;ENCdg ASCII;BN_FMT RI;BYT_OR MSB;WFID "D5, unknown coupling, 100.0us/div, 10000 points, Digitalmode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0</code> |
| <code>:CURVe?</code> | Returns the following values. Each value represents a data point:
<code>:CURVe 0,1,0,0,0,0,0,0,0</code> |

NOTE. You can also use the `WAVFrm?` query, which concatenates the `WFMOUtpre?` and `CURVe?` queries.

| WFMOutpre? Query results | Explanation |
|--|---|
| WFMOUTPRE:BYT_NR 1 | This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.) |
| BIT_NR 8 | This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly. |
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCDg command. (This value can also be set using the DATA:ENCDg command, which provides the ability to set the WFMOutpre:ENCDg , WFMOutpre:BN_Fmt,: and WFMOutpre:BYT_Or values using a single command.) |
| BN_FMT RI | This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCII encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCII encoding. |
| WFID "D5, unknown coupling, 100.0us/div, 10000 points, Digital mode" | This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFID? query. It cannot be changed. |
| NR_PT 25 | This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:STARt and DATA:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.) |
| PT_ORDER LINEAR | This value is always LINEar . |
| XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.) |
| XINCR 100.0000E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.) |
| XZERO 155.000E-12 | This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit . This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.) |
| PT_OFF 5000 | This value indicates the data point immediately following the trigger point relative to DATA:STARt in the waveform record.(If you would like to determine only this value, use the WFMOutpre:PT_Off? query.) |

| WFMOUtpre? Query results | Explanation |
|--------------------------|--|
| YUNIT "State" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YUNit? query.) |
| YMULT 1.0000 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the <code>WFMOUtpre:YUNit</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YMUlt? query.) |
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YOFF? query.) |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the <code>WFMOUtpre:YUNit</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YZero? query.) |

Example 3: Digital with 4 Bytes Per Point and Zoom Off

Goal: Transfer 25 points of Digital data from the instrument to a PC using 4 bytes per point and Zoom off.

| Command | Comment |
|-------------------------------------|--|
| <code>:DATA:SOURce CH1_D1</code> | |
| <code>:ACQuire:MAGnivu 0</code> | |
| <code>:DATA:START 1</code> | |
| <code>:DATA:STOP 25</code> | |
| <code>:WFMOUtpre:ENCdg ASCii</code> | |
| <code>:WFMOUtpre:BYT_Nr 4</code> | |
| <code>:HEADer 1</code> | |
| <code>:VERBose 1</code> | |
| <code>:WFMOUtpre?</code> | Returns the following values. Each value represents the current settings that a <code>CURVe?</code> query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):

<code>:WFMOUtpre:BYT_Nr 4;BIT_Nr 32;ENCdg ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0</code> |
| <code>:CURVe?</code> | Returns the following values. Each value represents a data point:

<code>:CURVe
FB386,FB366,FB3E6,FB366,FB3E6,FB32E,FB3A2,FB32E,FB3AA,FB366,FB3EA,FB366,FB3E2,FB36,FB3E6,FB366,FB3E6,FB346,FB3C6,FB346,FB3C6,FB34E,FB3C2,FB34E,FB3CA</code> |

NOTE. The returned hexadecimal data values are formatted without leading zeroes. For example, a 4-byte digital value of FB386 should be interpreted as 000FB386.

NOTE. You can also use the *WAVFrm?* query, which concatenates the *WFMOupre?* and *CURVe?* queries.

| WFMOupre? Query results | Explanation |
|---|--|
| WFMOUPRE:BYT_NR 4 | This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOupre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.) |
| BIT_NR 32 | This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOupre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly. |
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOupre:ENCDg command. (This value can also be set using the DATA:ENCDg command, which provides the ability to set the WFMOupre:ENCDg , WFMOupre:BN_Fmt , and WFMOupre:BYT_Or values using a single command.) |
| BN_FMT RI | This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOupre:BN_Fmt command. Note: this field is not applicable for ASCII encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOupre:BYT_Or command. Note: this field is not applicable for ASCII encoding. |
| WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode" | This quoted string represents information about the source waveform that would be returned by a WFMOupre:WFID? query. It cannot be changed. |
| NR_PT 25 | This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOupre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOupre:PT_Fmt? query.) |
| PT_ORDER LINEAR | This value is always LINEar. |
| XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. For live channels, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOupre:XUNIT? query.) |

| WFMOUtpre? Query results | Explanation |
|--------------------------|--|
| XINCR 100.0000E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOUtpre:XINcr? query.) |
| XZERO 155.000E-12 | This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOUtpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOUtpre:XZEro? query.) |
| PT_OFF 5000 | This value indicates the data point immediately following the trigger point relative to DATA:STARt in the waveform record. (If you would like to determine only this value, use the WFMOUtpre:PT_Off? query.) |
| YUNIT "State" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YUNit? query.) |
| YMULT 1.0000 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUtpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YMUIt? query.) |
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YOff? query.) |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the WFMOUtpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YZEro? query.) |

Example 4: Digital with 8 Bytes Per Point and Zoom Off

Goal: Transfer 25 points of Digital data from the instrument to a PC using 8 bytes per point and Zoom off.

| Command | Comment |
|------------------------|---------|
| :DATA:SOURce CH1_D1 | |
| :ACQuire:MAGnivu 0 | |
| :DATA:STARt 1 | |
| :DATA:STOP 25 | |
| :WFMOUtpre:ENCdg ASCii | |
| :WFMOUtpre:BYT_Nr 8 | |
| :HEADer 1 | |
| :VERBose 1 | |

| Command | Comment |
|------------|--|
| :WFMOupre? | Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):
<pre>:WFMOUPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0</pre> |
| :CURVe? | Returns the following values. Each value represents a data point:
<pre>:CURVe 80000FB386,E0000FB386,80000FB3E6,80000FB3E6,80000FB3E6, C8000FB3A6,8C000FB3A6,8C000FB3A6,84000FB3AE,CC000FB3A6, 8C000FB3E6,8C000FB3E6,84000FB3E6,80000FB3E6,80000FB3E6, 80000FB3E6,80000FB3E6,A0000FB3C6,80000FB3C6,80000FB3C6, 80000FB3C6,88000FB3C6,8C000FB3C6,8C000FB3C6,84000FB3CE</pre> |

NOTE. The returned hexadecimal data values are formatted without leading zeroes.

NOTE. You can also use the *WAVFrm?* query, which concatenates the *WFMOupre?* and *CURVe?* queries.

| WFMOupre? Query results | Explanation |
|-------------------------|--|
| WFMOUPRE:BYT_NR 8 | This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOupre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.) |
| BIT_NR 64 | This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOupre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly. |
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOupre:ENCdg command. (This value can also be set using the DATA:ENCdg command, which provides the ability to set the WFMOupre:ENCdg, WFMOupre:BN_Fmt, and WFMOupre:BYT_Or values using a single command.) |
| BN_FMT RI | This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOupre:BN_Fmt command. Note: this field is not applicable for ASCII encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOupre:BYT_Or command. Note: this field is not applicable for ASCII encoding. |

| WFMOUtpre? Query results | Explanation |
|---|--|
| WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode" | This quoted string represents information about the source waveform that would be returned by a <code>WFMOUtpre:WFID?</code> query. It cannot be changed. |
| NR_PT 25 | This value indicates the number of data points in the waveform record to be transferred using the <code>CURVe?</code> query. (If you would like to determine only this value, use the WFMOUtpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOUtpre:PT_Fmt? query.) |
| PT_ORDER LINEAR | This value is always <code>LINEAR</code> . |
| XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOUtpre:XUNit? query.) |
| XINCR 100.0000E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOUtpre:XINcr? query.) |
| XZERO 155.000E-12 | This value indicates the sub-sample time between the trigger sample (designated by <code>PT_OFF</code>) and the occurrence of the actual trigger in the waveform record. The value is in units of <code>WFMOUtpre:XUNit</code> . This is query only. (If you would like to determine only this value, use the WFMOUtpre:XZEro? query.) |
| PT_OFF 5000 | This value indicates the data point immediately following the trigger point relative to <code>DATA:START</code> in the waveform record. (If you would like to determine only this value, use the WFMOUtpre:PT_Off? query.) |
| YUNIT "State" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YUNit? query.) |
| YMULT 1.0000 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the <code>WFMOUtpre:YUNit</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YMUlt? query.) |
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YOff? query.) |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the <code>WFMOUtpre:YUNit</code> command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YZEro? query.) |

Example 5: Digital with 4 Bytes Per Point and Zoom On

Goal: Transfer 25 points of Digital data from the instrument to a PC using 4 bytes per point and Zoom on.

| Command | Comment |
|-----------------------|---|
| :DATA:SOURce CH1_D1 | |
| :ACQuire:MAGnivu 1 | |
| :DATA:START 1 | |
| :DATA:STOP 25 | |
| :WFMOupre:ENCdg ASCii | |
| :WFMOupre:BYT_Nr 4 | |
| :HEADer 1 | |
| :VERBoSe 1 | |
| :WFMOupre? | Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):

:WFMOUPRE:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 1.2121E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0 |
| :CURVe? | Returns the following values. Each value represents a data point:

:CURVE
FB6E6, FB666, FB6E6, FB666, FB666, FB6E6, FB666, FB6E6, FB666, FB666, FB6E6, FB666, FB6E6, FB666, FB666, FB6E6, FB666, FB666, FB6E6, FB666, FB666 |

NOTE. The returned hexadecimal data values are formatted without leading zeroes.

NOTE. You can also use the *WAVFrm?* query, which concatenates the *WFMOupre?* and *CURVe?* queries.

| WFMOupre? Query results | Explanation |
|-------------------------|---|
| WFMOUPRE:BYT_NR 4 | This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOupre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATA:WIDTH command.) |
| BIT_NR 32 | This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOupre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly. |

| WFMOUtpre? Query results | Explanation |
|---|---|
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOUtpre:ENCDg command. (This value can also be set using the DATA:ENCDg command, which provides the ability to set the WFMOUtpre:ENCDg , WFMOUtpre:BN_Fmt , and WFMOUtpre:BYT_Or values using a single command.) |
| BN_FMT RI | This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOUtpre:BN_Fmt command. Note: this field is not applicable for ASCII encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOUtpre:BYT_Or command. Note: this field is not applicable for ASCII encoding. |
| WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode" | This quoted string represents information about the source waveform that would be returned by a WFMOUtpre:WFID? query. It cannot be changed. |
| NR_PT 25 | This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOUtpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATA:START and DATA:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOUtpre:PT_Fmt? query.) |
| PT_ORDER LINEAR | This value is always LINEAR. |
| XUNIT "s" | This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is "s", representing seconds. When using the math waveform as a source, the value can be "s" or "Hz". This is query only. (If you would like to determine only this value, use the WFMOUtpre:XUNit? query.) |
| XINCR 1.2121E-9 | This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOUtpre:XINcr? query.) |
| XZERO 155.000E-12 | This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOUtpre:XUNit . This is query only. (If you would like to determine only this value, use the WFMOUtpre:XZEro? query.) |
| PT_OFF 5000 | This value indicates the data point immediately following the trigger point relative to DATA:START in the waveform record. (If you would like to determine only this value, use the WFMOUtpre:PT_Off? query.) |
| YUNIT "State" | This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YUNit? query.) |
| YMULT 1.0000 | This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUtpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOUtpre:YMUlt? query.) |

| WFMOupre? Query results | Explanation |
|-------------------------|---|
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOupre:YOFF? query.) |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform in units specified by the <code>WFMOupre:YUNIT</code> command. This is query only. (If you would like to determine only this value, use the WFMOupre:YZero? query.) |

Example 6: RF Frequency Domain Waveform

Goal: Transfer 5 points of RF frequency domain data from the instrument to a PC using 8 bytes per point.

NOTE. A frequency domain waveform used in data transfer from the instrument to a PC or other device may be one of the four RF frequency domain traces or the Spectrum Math waveform.

| Command | Comment |
|----------------------------|---|
| :DATA:SOURce CH1_SV_NORMal | |
| :DATA:STARt 495 | |
| :DATA:STOP 505 | |
| :WFMOupre:ENCDg ASCii | |
| :WFMOupre:BYT_Nr 8 | |
| :HEADer 1 | |
| :VERBose 1 | |
| :WFMOupre? | Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations):
:WFMOUPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT FP;BYT_OR MSB;WFID "RF_NORMAL, unknown coupling, 10.00w/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mw";NR_PT 21;PT_FMT Y;PT_ORDER LINEAR;XUNIT "Hz";XINCR 2.0000E+6;XZERO 1.4000E+9;PT_OFF 0;YUNIT "w";YMULT 1.5625E-3;YOFF 0.0E+0;YZERO 0.0E+0;DOMAIN FREQUENCY;WFMTYPE RF_FD;CENTERFREQUENCY 2.4000E+9;SPAN 2.0000E+9;REFLEVEL 1.0000E-3 |
| :CURVe? | Returns the following values. Each value represents a data point:
:CURVE
2.43108E-10,1.62648E-10,2.78478E-10,4.15163E-10,7.66223E-10,7.63905E-10,3.68375E-10,3.42668E-10,3.06422E-10,1.81561E-10,2.0223E-10,4.24327E-10,1.95298E-10,2.1304E-10,9.41791E-11,4.56513E-10,4.33067E-10,6.57215E-11,1.65388E-10,9.09116E-10,5.39507E-10 |

| WFMOupre? Query results | Explanation |
|---|--|
| BYT_NR 8 | This value specifies the number of bytes per data point in the waveform data. This value is fixed for RF frequency domain traces. |
| BIT_NR 64 | This value specifies the number of bits per data point in the waveform data. This value is fixed for RF frequency domain traces. |
| ENCDG ASCII | This value specifies the encoding of the waveform data. To change this value (the other possibility for RF frequency domain traces is FPBINARY), use the WFMOupre:ENCDg command. This value can also be set using the DATa:ENCDg command. |
| BN_FMT FP | This value specifies the binary format, which in this case is FP (floating point). This value is fixed for RF frequency domain traces. Note: This field is not applicable for ASCII encoding. |
| BYT_OR MSB | This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOupre:BYT_Or command. Note: This field is not applicable for ASCII encoding. |
| WFID "CH1_SV_NORMal, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW" | This quoted string represents information about the source waveform that would be returned by a WFMOupre:WFId? Query. It cannot be changed. |
| NR_PT 21 | This value indicates the number of data points in the waveform record. (If you would like to determine only this value, use the WFMOupre:NR_Pt? query.) Note: This value is typically equal to the full record length of the RF frequency domain trace, but you also have the option to transfer only a portion of the record using the DATs:START and DATa:STOP commands. |
| PT_FMT Y | This value indicates the format of the data points in the waveform record. In this case, the value represents YF (Amplitude vs. Frequency) format. This is query only. (If you would like to determine only this value, use the WFMOupre:PT_Fmt? query.) |
| PT_ORDER LINEAR . | This value is always LINear |
| XUNIT "Hz" | This value indicates the units of the x-axis of the waveform record. This value is always "Hz" for RF frequency domain traces. This is query only. (If you would like to determine only this value, use the WFMOupre:XUNit? query.) |
| XINCR 2.0000E+6 | This value indicates the frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOupre:XINcr? query.) |
| XZERO 1.4000E+9 | This value indicates the frequency, in hertz, of the first data point in the waveform record. This frequency is relative to the time of the trigger, which is always 0. This XZEro frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOupre:XZEro? query.) |
| PT_OFF 0 | This is a query provided only for compatibility with other Tektronix instruments. The returned value is always 0. (If you would like to determine only this value, use the WFMOupre:PT_Off? query.) |
| YUNIT "W" | This value indicates the units of data points in the waveform record. This value depends on the vertical units of the source waveform – in this case, watts. This is query only. (If you would like to determine only this value, use the WFMOupre:YUNit? query.) |
| YMULT 1.5625E-3 | This value indicates the multiplying factor to convert the data point values to the units specified by the:WFMOupre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOupre:YMUlt? query.) |

| | |
|---------------------------|--|
| YOFF 0.0E+0 | This value indicates the vertical position of the source waveform. This value is unused for RF frequency domain traces and is always 0. |
| YZERO 0.0E+0 | This value indicates the vertical offset of the source waveform. This value is unused for RF frequency domain traces and is always 0. |
| DOMAIN FREQUENCY | This value indicates the domain in which the source waveform is displayed and stored. For RF time domain traces, the domain is Time and waveform transfer information is treated as integer information. For RF frequency domain traces, the domain is Frequency and waveform transfer information is treated as floating point information. This is query only. (If you would like to determine only this value, use the WFMOupre:DOMain? query.) |
| WFMTYPE RF_FD | This value indicates the type of the source waveform. RF_FD indicates an RF frequency domain trace (frequency domain waveform). This is query only. (If you would like to determine only this value, use the WFMOupre:WFMTYPE? query.) |
| CENTERFREQUENCY 2.4000E+9 | This value indicates the center frequency, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOupre:CENTERFREQuency? query.) |
| SPAN 2.0000E+9 | This value indicates the frequency span, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOupre:SPAN? query.) |
| REFLEVEL 1.0000E-3 | This value indicates the reference level, in watts, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOupre:REFLEvel? query.) |

Example 7: Baseband IQ Waveform

Goal: Transfer Baseband I and Q data from the instrument to a PC.

NOTE. Requires SV-RFVT option.

| Command | Comment |
|------------------------------|---|
| :DATA:SOURce CH1_BASEBAND_IQ | |
| :DATA:WIDth 2 | Set data width for every I,Q pair to 2. |
| :WFMOupre:ENCdg SRIBinary | Request data LSB first. |
| :WFMOupre:YMUlt? | Apply 2*YMUlt scaling to I and Q samples returned by CURVe? |

| Command | Comment |
|-------------|--|
| :WFMOutpre? | <p>Returns the following values. Each value represents the current settings that aCURVe? query will use to format the data that will be transferred from the instrument to a PC or other device:</p> <pre data-bbox="634 373 1458 667">:WFMOUTPRE:BYT_NR 1;BIT_NR 8;:WFMOUTPRE:ENCDG BINARY;BN_FMT RI;:WFMOUTPRE:ASC_FMT INTEGER;BYT_OR MSB;:WFMOUTPRE:WFID "CH1_SV_BASEBAND_IQ, 5062 points, Center Freq: 1GHz, Span: 312.5MHz";NR_PT 9;:WFMOUTPRE:PT_FMT ENV;PT_ORDER LINEAR;:WFMOUTPRE:XUNIT "s";XINCR 1.280E-9;:WFMOUTPRE:XZERO 815.000E-12;PT_OFF 0;:WFMOUTPRE:YUNIT "V";YMULT 8.0000E-3;:WFMOUTPRE:YOFF 0.0E+0;YZERO0.0E+0;:WFMOUTPRE:DOMAIN TIME;WFMTYPE F_TD;:WFMOUTPRE:CENTERFREQUENCY 1.0000E+9;SPAN 312.5000E+6</pre> |
| :CURVe? | <p>CURVe? will transmit NR_Pt points which are interleaved I and Q pairs. The result can be read from the binary buffer as uint16 to get I and Q pair values in "digitizing levels". Multiply the digitizing level value by 2*YMULT to get absolute I and Q point values.</p> |

Appendix E: Search and Trigger Command Sequence Examples

The following are some example command sequences that show a variety of different searches and triggers. The commands in these sequences are not order-dependent.

To use these examples, connect channel 1, channel 2, channel 3 and channel 4 to the probe compensation signal located on the right hand side of the front panel.

The search and trigger command group sections contain more information on general search and trigger concepts. ,

Example 1: Single Threshold Edge Search

Goal: Search the channel 2 waveform and place a mark at each instance where it crosses below a threshold of 1.4 volts.

| Command | Comment |
|---|--|
| *RST;:*OPC? | Resets the instrument and waits for that operation to complete (approximately 1 to 5 seconds depending on the complexity of the previous setup). |
| display:waveview1:ch2:state 1 | Turns the CH2 waveform on. |
| :AUTOset EXECute>:*OPC? | Autosets the displayed waveform CH2 and waits for the auto setup to complete. |
| SEARCH:SEARCH1:TRIGger:A:TYPE EDGE | Specifies that this will be an edge search (a mark will be placed when the source waveform passes through a specified threshold level in the specified direction). |
| SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH2 | Specifies the CH2 waveform as the source waveform. |
| SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH2 1.4 | Specifies 1.4 volts as the threshold level. |
| SEARCH:SEARCH1:TRIGger:A:EDGE:SLOpe FALL | Specifies the falling edge as the direction. |
| SEARCH:SEARCH1:STATE 1 | Turns the search on. |
| SEARCH:SEARCH1:TOTal? | Returns 4, indicating that CH2 fell below the 1.4 volt threshold 4 times. |

Example 2: Single Threshold Edge Trigger

Goal: Trigger on the channel 2 waveform when the waveform crosses below a threshold of 1.4 volts.

| Command | Comment |
|--------------------------------|--|
| *RST | Resets the instrument. Wait for the reset to complete (approximately 3 seconds). |
| :DISPLAY:WAVEVIEW1:CH2:STATE 1 | Turns the CH2 waveform on. |

| Command | Comment |
|-----------------------------------|--|
| :TRIGger:A:TYPe EDGE | Specifies that this will be an edge trigger (trigger will occur when the source waveform passes through a specified threshold level in the specified direction). |
| :TRIGger:A:EDGE:SOUrce CH2 | Specifies the CH2 waveform as the source waveform. |
| :TRIGger:A:LOWerthreshold:CH2 1.4 | Specifies 1.4 volts as the threshold level. |
| :TRIGger:A:EDGE:SLOpe FALL | Specifies as falling edge as the direction. |
| :TRIGger:STATE? | Should return TRIGGER, not AUTO. |

Example 3: Dual Threshold Runt Search

Goal: Search the channel 3 waveform for negative runt pulses and place a mark at each instance when the waveform drops below an upper threshold of 1.4 volts, but does not cross a lower threshold of -2 volts before re-crossing the upper threshold. The pulse width must be less than 600 microseconds.

| Command | Comment |
|--|---|
| *RST | Resets the instrument. Wait for the reset to complete (approximately 3 seconds). |
| :DISPLAY:WAVEVIEW1:CH3:STATE 1 | Turns the CH3 waveform on. |
| :AUTOset EXECute | Autosets the displayed waveform CH3. Wait for the autoset to complete (approximately 3 seconds). |
| SEARCH:SEARCH1:TRIGger:A:TYPe RUNT | Specifies that this will be a runt search (a mark will be placed on a pulse amplitude that crosses one threshold but fails to cross a second threshold before re-crossing the first). |
| SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH3 | Specifies to use channel 3 as the source waveform. |
| SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH3 -2 | Specifies to use -2 volts as the lower threshold. |
| SEARCH:SEARCH1:TRIGger:A:UPPerthreshold: CH3 1.4 | Specifies to use 1.4 volts as the upper threshold. |
| SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGative | Specifies to search for when the runt polarity is negative. |
| SEARCH:SEARCH1:TRIGger:A:RUNT:WIDth 600E-6 | Specifies a pulse width of 600E-6 seconds. |
| SEARCH:SEARCH1:TRIGger:A:RUNT:WHEn LESSthan | Specifies to search for when the pulse width is less than 600E-6 seconds. |
| SEARCH:SEARCH1:STATE 1 | Turns the search on. |
| SEARCH:SEARCH1:TOTal? | Returns 3, indicating that 3 total negative runt pulses were less than 600E-6 seconds wide |

NOTE. You could use a similar command sequence with a transition type search.

Example 4: Single Threshold Logic Search on Three Waveforms

Goal: Search the channel 1, 2 and 3 waveforms and place a mark at each instance when either channel 1 is above 1.4 volts, channel 2 is above 1.5 volts, or channel 3 is above 1.3 volts.

| Command | Comment |
|--|---|
| *RST | Resets the instrument. Wait for the reset to complete (approximately 3 seconds). |
| :DISPLAY:WAVEVIEW1:CH1:STATE 1 | Turns the CH1 waveform on. |
| :DISPLAY:WAVEVIEW1:CH2:STATE 1 | Turns the CH2 waveform on. |
| :DISPLAY:WAVEVIEW1:CH3:STATE 1 | Turns the CH3 waveform on. |
| :AUTOset EXECute | Autosets the displayed waveforms CH1, CH2 and CH3. Wait for the autose to complete (approximately 3 seconds). |
| SEARCH:SEARCH1:TRIGger:A:TYPE LOGic | Specifies that this will be a logic search (a mark will be placed when all channels transition to the specified state). |
| SEARCH:SEARCH1:TRIGger:A:LOGic:INPut:CH1 HIGH | Specifies the Boolean logic criteria for channel 1; in this case, high. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:INPut:CH2 HIGH | Specifies the Boolean logic criteria for channel 2; in this case, high. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:INPut:CH3 HIGH | Specifies the Boolean logic criteria for channel 3; in this case, high. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:THReshold:CH1 1.4 | Specifies to use 1.4 volts as the threshold for CH1. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:THReshold:CH2 1.5 | Specifies to use 1.5 volts as the threshold for CH2. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:THReshold:CH3 1.3 | Specifies to use 1.3 volts as the threshold for CH3. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:PATtern:WHEn TRUE | Specifies the condition for generating a logic pattern search; in this case, true. |
| SEARCH:SEARCH1:TRIGger:A:LOGic:FUNCTION OR | Specifies the logic operator for the logic search; in this case, OR. |
| SEARCH:SEARCH1:STATE 1 | Turns the search on. |
| SEARCH:SEARCH1:TOTal? | Returns 3 or 4, indicating 3 or 4 times when one of channels 1, 2 or 3 became high.

Note: Depending upon the number of transitions displayed, you may get 3 or 4 search marks for this example. If you get 3 search marks, try adjusting the horizontal position until you see 4 search marks. |

Glossary

ASCII

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

Address

A 7-bit code that identifies an instrument on the communication bus. The digitizing instrument must have a unique address for the controller to recognize and transmit commands to it.

Backus-Naur Form (BNF)

A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller

A computer or other device that sends commands to and accepts responses from the digitizing instrument.

EOI

A mnemonic referring to the control line End or Identify. One of the two possible end-of-message terminators.

EOM

A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

Equivalent-Time sampling (ET)

A sampling mode in which the instrument acquires signals over many repetitions of the event. This instrument uses a type of equivalent time sampling called random equivalent time sampling. It utilizes an internal clock that runs asynchronously with respect to the input signal and the signal trigger. The instrument takes samples continuously, independent of the trigger position, and displays them based on the time difference between the sample and the trigger. Although the samples are taken sequentially in time, they are random with respect to the trigger.

Real-Time sampling

A sampling mode where the instrument samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

IEEE

An acronym for the Institute for Electrical and Electronic Engineers.

Serial poll

A device on the bus can request service from the Controller by asserting the SRQ line. When a controller acknowledges the SRQ, it serial polls each device on the bus to determine which device on the bus requested service. Any device requesting service returns a status byte indicating it needs to be serviced and then

unasserts the SRQ line. Devices not requiring service return a status byte that indicates they do not need servicing.

TEKSecure

A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.

Index

A

- ACQuire?, 2-73
- ACQuire:MAXSamplerate?, 2-73
- ACQuire:MODE, 2-73
- ACQuire:NUMACq?, 2-74
- ACQuire:NUMAVg, 2-75
- ACQuire:SEQuence:CURrent?, 2-75
- ACQuire:SEQuence:MODE, 2-76
- ACQuire:SEQuence:NUMSEQuence, 2-76
- ACQuire:STATE, 2-76
- ACQuire:STOPAfter, 2-77
- ACTONEVent:ENable, 2-78
- ACTONEVent:LIMit, 2-78
- ACTONEVent:LIMITCount, 2-79
- ACTONEVent:MASKFail:ACTION:SAVEIMAGE:
STATE, 2-79
- ACTONEVent:MASKFail:ACTION:
SAVEWAVEform:STATE, 2-80
- ACTONEVent:MASKFail:ACTION:SRQ:
STATE, 2-81
- ACTONEVent:MASKFail:ACTION:STOPACQ:
STATE, 2-81
- ACTONEVent:MASKHit:ACTION:SAVEIMAGE:
STATE, 2-82
- ACTONEVent:MASKHit:ACTION:
SAVEWAVEform:STATE, 2-83
- ACTONEVent:MASKHit:ACTION:SRQ:
STATE, 2-83
- ACTONEVent:MASKHit:ACTION:STOPACQ:
STATE, 2-84
- ACTONEVent:MASKPass:ACTION:SAVEIMAGE:
STATE, 2-84
- ACTONEVent:MASKPass:ACTION:
SAVEWAVEform:STATE, 2-85
- ACTONEVent:MASKPass:ACTION:SRQ:
STATE, 2-86
- ACTONEVent:MASKPass:ACTION:STOPACQ:
STATE, 2-86
- ACTONEVent:MEASUrement:ACTION:
SAVEIMAGE:STATE, 2-87
- ACTONEVent:MEASUrement:ACTION:
SAVEWAVEform:STATE, 2-88
- ACTONEVent:MEASUrement:ACTION:SRQ:
STATE, 2-89
- ACTONEVent:MEASUrement:ACTION:STOPACQ:
STATE, 2-89
- ACTONEVent:SEARCH:ACTION:SAVEIMAGE:
STATE, 2-90
- ACTONEVent:SEARCH:ACTION:
SAVEWAVEform:STATE, 2-91
- ACTONEVent:SEARCH:ACTION:SRQ:
STATE, 2-91
- ACTONEVent:SEARCH:ACTION:STOPACQ:
STATE, 2-92
- ACTONEVent:TRIGger:ACTION:SAVEIMAGE:
STATE, 2-93
- ACTONEVent:TRIGger:ACTION:SAVEWAVEform:
STATE, 2-93
- ACTONEVent:TRIGger:ACTION:SRQ:STATE, 2-94
- ACTONEVent:TRIGger:ACTION:STOPACQ:
STATE, 2-95
- AFG:AMPLitude, 2-95
- AFG:ARBitrary:SOUrce, 2-96
- AFG:BURSt:CCOUnt, 2-96
- AFG:BURSt:TRIGger, 2-97
- AFG:FREQuency, 2-97
- AFG:FUNcTION, 2-98
- AFG:HIGHLevel, 2-98
- AFG:LOWLevel, 2-99
- AFG:NOISEAdd:PERCent, 2-99
- AFG:NOISEAdd:STATE, 2-100
- AFG:OFFSet, 2-100
- AFG:OUTPut:LOAd:IMPEDance, 2-101
- AFG:OUTPut:MODE, 2-101
- AFG:OUTPut:STATE, 2-102
- AFG:PERIod, 2-102
- AFG:PULse:WIDth, 2-103
- AFG:RAMP:SYMmetry, 2-103
- AFG:SQUare:DUty, 2-104
- ALias, 2-104
- ALias:CATalog?, 2-105
- ALias:DEFine, 2-105
- ALias:DELEte, 2-106
- ALias:DELEte:ALL, 2-107
- ALias:DELEte:NAME, 2-107
- ALias:STATE, 2-107
- ALLEv?, 2-108
- AUTOSAVEPITIMEOUT, 2-108

AUTOSAVEUITIMEOUT, 2-109
 AUTOSet, 2-109
 AUTOSet:ACQuisition:ENable, 2-110
 AUTOSet:ENable, 2-110
 AUTOSet:HORizontal:ENable, 2-111
 AUTOSet:TRIGger:ENable, 2-111
 AUTOSet:VERTical:ENable, 2-112
 AUTOSet:VERTical:OPTIMize, 2-113
 AUXout:EDGE, 2-113
 AUXout:SOURce, 2-114

B

BATTery:ACPOWer?, 2-114
 BATTery:SLOT<1,2>:CHARGE?, 2-115
 BATTery:SLOT<1,2>:INSTalled?, 2-115
 BATTery:SLOT<1,2>:SERIALnumber?, 2-116
 BATTery:SLOT<1,2>:TIMETOEMPty?, 2-116
 BATTery:SLOT<1,2>:TIMETOFULL?, 2-116
 BUS:ADDNew, 2-117
 BUS:B<x>:CAN:BITRate, 2-117
 BUS:B<x>:CAN:BITRate:VALue, 2-118
 BUS:B<x>:CAN:FD:BITRate, 2-119
 BUS:B<x>:CAN:FD:BITRate:CUSTom, 2-119
 BUS:B<x>:CAN:SAMPLEpoint, 2-120
 BUS:B<x>:CAN:SIGNAL, 2-121
 BUS:B<x>:CAN:SOURce, 2-121
 BUS:B<x>:CAN:STANDard, 2-122
 BUS:B<x>:CAN:THReshold, 2-122
 BUS:B<x>:DISplay:FORMat, 2-123
 BUS:B<x>:DISplay:LAYout, 2-124
 BUS:B<x>:I2C:CLOCK:SOURce, 2-124
 BUS:B<x>:I2C:CLOCK:THReshold, 2-125
 BUS:B<x>:I2C:DATA:SOURce, 2-126
 BUS:B<x>:I2C:DATA:THReshold, 2-126
 BUS:B<x>:I2C:RWINADDR, 2-127
 BUS:B<x>:LABel:COLor, 2-128
 BUS:B<x>:LABel:FONT:BOLD, 2-128
 BUS:B<x>:LABel:FONT:ITALic, 2-129
 BUS:B<x>:LABel:FONT:SIZE, 2-129
 BUS:B<x>:LABel:FONT:TYPE, 2-130
 BUS:B<x>:LABel:FONT:UNDERline, 2-130
 BUS:B<x>:LABel:name, 2-131
 BUS:B<x>:LABel:XPOS, 2-131
 BUS:B<x>:LABel:YPOS, 2-132
 BUS:B<x>:LIN:BITRate, 2-132
 BUS:B<x>:LIN:BITRate:CUSTom, 2-133
 BUS:B<x>:LIN:IDFORmat, 2-134

BUS:B<x>:LIN:POLarity, 2-134
 BUS:B<x>:LIN:SAMPLEpoint, 2-135
 BUS:B<x>:LIN:SOURce, 2-135
 BUS:B<x>:LIN:SOURce:THReshold, 2-136
 BUS:B<x>:LIN:STANDard, 2-136
 BUS:B<x>:PARallel:ALLTHResholds, 2-137
 BUS:B<x>:PARallel:ALLTHResholds:APPLY, 2-138
 BUS:B<x>:PARallel:BIT<n>SOURce:
 THReshold, 2-139
 BUS:B<x>:PARallel:BIT<x>SOURce, 2-138
 BUS:B<x>:PARallel:CLOCK:EDGE, 2-139
 BUS:B<x>:PARallel:CLOCK:ISCLOCKED, 2-140
 BUS:B<x>:PARallel:CLOCKSOURce, 2-140
 BUS:B<x>:PARallel:CLOCKSOURce:
 THReshold, 2-141
 BUS:B<x>:RS232C:BITRate, 2-142
 BUS:B<x>:RS232C:BITRate:CUSTom, 2-142
 BUS:B<x>:RS232C:DATAbits, 2-143
 BUS:B<x>:RS232C:DELIMiter, 2-144
 BUS:B<x>:RS232C:DISplaymode, 2-144
 BUS:B<x>:RS232C:PARity, 2-145
 BUS:B<x>:RS232C:POLarity, 2-146
 BUS:B<x>:RS232C:SOURce, 2-146
 BUS:B<x>:RS232C:SOURce:THReshold, 2-147
 BUS:B<x>:SENT:CHANWidth, 2-147
 BUS:B<x>:SENT:NIBBLECount, 2-148
 BUS:B<x>:SENT:NUMCHANnel, 2-149
 BUS:B<x>:SENT:PAUSEPULSe, 2-149
 BUS:B<x>:SENT:POLARITY, 2-150
 BUS:B<x>:SENT:SLOW, 2-150
 BUS:B<x>:SENT:SOURce, 2-151
 BUS:B<x>:SENT:THRESHold, 2-152
 BUS:B<x>:SENT:TICKTIME, 2-152
 BUS:B<x>:SENT:TICKTOLerance, 2-153
 BUS:B<x>:SPI:BITOrder, 2-153
 BUS:B<x>:SPI:CLOCK:POLarity, 2-154
 BUS:B<x>:SPI:CLOCK:SOURce, 2-155
 BUS:B<x>:SPI:CLOCK:THReshold, 2-156
 BUS:B<x>:SPI:DATA:POLarity, 2-156
 BUS:B<x>:SPI:DATA:SIZE, 2-157
 BUS:B<x>:SPI:DATA:SOURce, 2-157
 BUS:B<x>:SPI:DATA:THReshold, 2-158
 BUS:B<x>:SPI:FRAMING, 2-158
 BUS:B<x>:SPI:IDLETime, 2-159
 BUS:B<x>:SPI:MISo:DATA:POLarity, 2-160
 BUS:B<x>:SPI:MISo:INPut, 2-160
 BUS:B<x>:SPI:MISo:THReshold, 2-161

BUS:B<x>:SPI:MOSi:DATA:POLarity, 2-161
 BUS:B<x>:SPI:MOSi:INPut, 2-162
 BUS:B<x>:SPI:MOSi:THReshold, 2-163
 BUS:B<x>:SPI:NUMBer:INputs, 2-163
 BUS:B<x>:SPI:SElect:POLarity, 2-164
 BUS:B<x>:SPI:SElect:SOUrce, 2-164
 BUS:B<x>:SPI:SElect:THReshold, 2-165
 BUS:B<x>:TYPE, 2-166
 BUS:DELeTe, 2-167
 BUS:LIST?, 2-167
 BUSTABLE:ADDNew, 2-167
 BUSTABLE:DELeTe, 2-168
 BUSTABLE:LIST?, 2-168
 BUSY?, 2-168

C

*CAL?, 2-169
 CALibrate?, 2-169
 CALibrate:INTERNaL, 2-170
 CALibrate:INTERNaL:STARt, 2-170
 CALibrate:INTERNaL:STATus?, 2-171
 CALibrate:PWRUpstatus?, 2-172
 CALLOUTS:CALLOUT<x>, 2-172
 CALLOUTS:CALLOUT<x>:BOOKMark:
 SOURCE, 2-172
 CALLOUTS:CALLOUT<x>:BOOKMark:
 XPOS, 2-173
 CALLOUTS:CALLOUT<x>:COLOR, 2-173
 CALLOUTS:CALLOUT<x>:DISPLAyPOSition:
 X, 2-174
 CALLOUTS:CALLOUT<x>:DISPLAyPOSition:
 Y, 2-174
 CALLOUTS:CALLOUT<x>:FONT:BOLD, 2-175
 CALLOUTS:CALLOUT<x>:FONT:ITALIC, 2-175
 CALLOUTS:CALLOUT<x>:FONT:SIZE, 2-176
 CALLOUTS:CALLOUT<x>:FONT:TYPE, 2-176
 CALLOUTS:CALLOUT<x>:FONT:
 UNDERLine, 2-176
 CALLOUTS:CALLOUT<x>:TEXT, 2-177
 CALLOUTS:CALLOUT<x>:TYPE, 2-177
 CH<x>?, 2-178
 CH<x>:BANdwidth, 2-178
 CH<x>:CLIPping?, 2-179
 CH<x>:COUPling, 2-179
 CH<x>:DESKew, 2-180
 CH<x>:DITHERrange, 2-181
 CH<x>:INVert, 2-181

CH<x>:LABel:COLor, 2-182
 CH<x>:LABel:FONT:BOLD, 2-182
 CH<x>:LABel:FONT:ITALic, 2-183
 CH<x>:LABel:FONT:SIZE, 2-183
 CH<x>:LABel:FONT:TYPE, 2-184
 CH<x>:LABel:FONT:UNDERline, 2-184
 CH<x>:LABel:NAME, 2-185
 CH<x>:LABel:XPOS, 2-185
 CH<x>:LABel:YPOS, 2-186
 CH<x>:OFFSet, 2-186
 CH<x>:POSition, 2-187
 CH<x>:PROBEFunc:EXTAtten, 2-187
 CH<x>:PROBEFunc:EXTDBatten, 2-188
 CH<x>:PROBEFunc:EXTUnits, 2-188
 CH<x>:PROBEFunc:EXTUnits:STATE, 2-189
 CH<x>:SCALE, 2-189
 CH<x>:SCALERATio, 2-190
 CH<x>:TERmination, 2-190
 CH<x>:VTERm:BIAS, 2-191
 CLEAR, 2-191
 *CLS, 2-192
 Command Groups, 2-11
 CONFIGuration:ANALog:BANdWidth?, 2-193
 CONNected:REQUested:STATus , 2-193
 CONNected:STATus?, 2-193
 CONNected:USAGe:TRack:REQUested:
 STATus, 2-194
 CONNected:USAGe:TRack:STATus?, 2-194
 CURVe?, 2-195
 CURVEStream?, 2-196

D

DATA, 2-198
 DATA:ENCdg, 2-199
 DATA:MODE , 2-201
 DATA:RESample, 2-201
 DATA:SOUrce, 2-202
 DATA:SOUrce:AVAILable?, 2-203
 DATA:STARt, 2-203
 DATA:STOP, 2-204
 DATA:WIDth, 2-205
 DATE?, 2-205
 DCH<x>_D<x>:LABel:COLor, 2-206
 DCH<x>_D<x>:LABel:FONT:BOLD, 2-207
 DCH<x>_D<x>:LABel:FONT:ITALic, 2-207
 DCH<x>_D<x>:LABel:FONT:SIZE, 2-208
 DCH<x>_D<x>:LABel:FONT:TYPE, 2-208

- DCH<x>_D<x>:LABel:FONT:UNDERline, 2-209
DCH<x>_D<x>:LABel:NAME, 2-209
DCH<x>_D<x>:THREshold, 2-206
*DDT, 2-210
DESE, 2-211
DIAG:LOOP:OPTion, 2-212
DIAG:LOOP:OPTion:NTIMes, 2-212
DIAG:LOOP:STOP, 2-212
DIAG:MODE, 2-213
DIAG:RESUlT?, 2-213
DIAG:RESUlT:FLAg?, 2-214
DIAG:RESUlT:LOG?, 2-214
DIAG:SELEct, 2-214
DIAG:STATE, 2-215
DISplay?, 2-215
DISplay:{CH<x>|Math<x>|REF<x>}:
 INVERTColor, 2-216
DISplay:{CH<x>|Math<x>|REF<x>}:
 NORMALColor, 2-217
DISplay:COLors, 2-217
DISplay:GLObal:B<x>:STATE, 2-218
DISplay:GLObal:CH<x>:STATE, 2-218
DISplay:GLObal:DCH<x>:STATE, 2-219
DISplay:GLObal:MATH<x>:STATE, 2-220
DISplay:GLObal:REF<x>:STATE, 2-220
DISplay:INTENSITy?, 2-221
DISplay:INTENSITy:BACKLighT, 2-221
DISplay:INTENSITy:BACKLighT:AUTODim:
 ENABle, 2-222
DISplay:INTENSITy:BACKLighT:AUTODim:
 TIME, 2-222
DISplay:MATHFFTView<n>:CURSor:
 ROLOCATION, 2-223
DISplay:MATHFFTView<x>:AUTOScale, 2-223
DISplay:MATHFFTView<x>:CURSor:
 ASOUrce?, 2-224
DISplay:MATHFFTView<x>:CURSor:
 BSOUrce?, 2-224
DISplay:MATHFFTView<x>:CURSor:DDT?, 2-225
DISplay:MATHFFTView<x>:CURSor:
 FUNCTion, 2-225
DISplay:MATHFFTView<x>:CURSor:HBArs:
 APOSition, 2-226
DISplay:MATHFFTView<x>:CURSor:HBArs:
 AUNIts?, 2-226
DISplay:MATHFFTView<x>:CURSor:HBArs:
 BPOSition, 2-227
DISplay:MATHFFTView<x>:CURSor:HBArs:
 BUNIts?, 2-227
DISplay:MATHFFTView<x>:CURSor:HBArs:
 DELTA?, 2-228
DISplay:MATHFFTView<x>:CURSor:MODE, 2-228
DISplay:MATHFFTView<x>:CURSor:
 ONEOVERDELTAVALUE?, 2-229
DISplay:MATHFFTView<x>:CURSor:SCREEN:
 AXPOSition, 2-229
DISplay:MATHFFTView<x>:CURSor:SCREEN:
 AYPOSition, 2-230
DISplay:MATHFFTView<x>:CURSor:SCREEN:
 BXPOSition, 2-230
DISplay:MATHFFTView<x>:CURSor:SCREEN:
 BYPOSition, 2-231
DISplay:MATHFFTView<x>:CURSor:STATE, 2-231
DISplay:MATHFFTView<x>:CURSor:VBArS:
 APOSition, 2-232
DISplay:MATHFFTView<x>:CURSor:VBArS:
 AUNIts?, 2-232
DISplay:MATHFFTView<x>:CURSor:VBArS:
 BPOSition, 2-233
DISplay:MATHFFTView<x>:CURSor:VBArS:
 BUNIts?, 2-233
DISplay:MATHFFTView<x>:CURSor:VBArS:
 DELTA?, 2-234
DISplay:MATHFFTView<x>:CURSor:WAVEform:
 APOSition, 2-234
DISplay:MATHFFTView<x>:CURSor:WAVEform:
 BPOSition, 2-234
DISplay:MATHFFTView<x>:GRIDlines, 2-235
DISplay:MATHFFTView<x>:MATH:MATH<x>:
 STATE, 2-236
DISplay:MATHFFTView<x>:XAXIS:SCALE, 2-236
DISplay:MATHFFTView<x>:YAXIS:SCALE, 2-237
DISplay:MATHFFTView<x>:ZOOM:XAXIS:
 FROM, 2-237
DISplay:MATHFFTView<x>:ZOOM:XAXIS:
 TO, 2-238
DISplay:MATHFFTView<x>:ZOOM:YAXIS:
 FROM, 2-238
DISplay:MATHFFTView<x>:ZOOM:YAXIS:
 TO, 2-239
DISplay:PERsistence, 2-239
DISplay:PERsistence:RESET, 2-240
DISplay:PLOTView<n>:CURSor:
 ROLOCATION, 2-240

- DISplay:PLOTView<x>:AUTOScale, 2-241
- DISplay:PLOTView<x>:CURSor:ASOUrce?, 2-242
- DISplay:PLOTView<x>:CURSor:BSOUrce?, 2-242
- DISplay:PLOTView<x>:CURSor:DDT?, 2-242
- DISplay:PLOTView<x>:CURSor:FUNcTION, 2-243
- DISplay:PLOTView<x>:CURSor:HBArs:
 - APOsition, 2-243
- DISplay:PLOTView<x>:CURSor:HBArs:
 - AUNIts?, 2-244
- DISplay:PLOTView<x>:CURSor:HBArs:
 - BPOsition, 2-244
- DISplay:PLOTView<x>:CURSor:HBArs:
 - BUNIts?, 2-245
- DISplay:PLOTView<x>:CURSor:HBArs:
 - DELTA?, 2-245
- DISplay:PLOTView<x>:CURSor:MODe, 2-246
- DISplay:PLOTView<x>:CURSor:
 - ONEOVERDELTAVALUE?, 2-246
- DISplay:PLOTView<x>:CURSor:SCREEN:
 - AXPOsition, 2-247
- DISplay:PLOTView<x>:CURSor:SCREEN:
 - AYPOsition, 2-247
- DISplay:PLOTView<x>:CURSor:SCREEN:
 - BXPOsition, 2-248
- DISplay:PLOTView<x>:CURSor:SCREEN:
 - BYPOsition, 2-248
- DISplay:PLOTView<x>:CURSor:
 - SPLITMODE, 2-249
- DISplay:PLOTView<x>:CURSor:STATE, 2-249
- DISplay:PLOTView<x>:CURSor:VBArS:
 - APOsition, 2-250
- DISplay:PLOTView<x>:CURSor:VBArS:
 - BPOsition, 2-250
- DISplay:PLOTView<x>:CURSor:VBArS:
 - DELTA?, 2-251
- DISplay:PLOTView<x>:CURSor:VBArS:
 - UNIts?, 2-251
- DISplay:PLOTView<x>:CURSor:WAVEform:
 - APOsition, 2-252
- DISplay:PLOTView<x>:CURSor:WAVEform:
 - BPOsition, 2-252
- DISplay:PLOTView<x>:GRIDlines, 2-253
- DISplay:PLOTVIEW<x>:XAXIS:SCALE, 2-254
- DISplay:PLOTVIEW<x>:YAXIS:SCALE, 2-254
- DISplay:PLOTView<x>:ZOOM:XAXIS:
 - FROM, 2-255
- DISplay:PLOTView<x>:ZOOM:XAXIS:TO, 2-255
- DISplay:PLOTView<x>:ZOOM:YAXIS:
 - FROM, 2-256
- DISplay:PLOTView<x>:ZOOM:YAXIS:TO, 2-256
- DISplay:REFFFTView<n>:CURSor:
 - ROLOCATION, 2-257
- DISplay:REFFFTView<x>:AUTOScale, 2-257
- DISplay:REFFFTView<x>:CURSor:
 - ASOUrce?, 2-258
- DISplay:REFFFTView<x>:CURSor:
 - BSOUrce?, 2-258
- DISplay:REFFFTView<x>:CURSor:DDT?, 2-259
- DISplay:REFFFTView<x>:CURSor:
 - FUNcTION, 2-259
- DISplay:REFFFTView<x>:CURSor:HBArs:
 - APOsition, 2-260
- DISplay:REFFFTView<x>:CURSor:HBArs:
 - AUNIts?, 2-260
- DISplay:REFFFTView<x>:CURSor:HBArs:
 - BPOsition, 2-261
- DISplay:REFFFTView<x>:CURSor:HBArs:
 - BUNIts?, 2-261
- DISplay:REFFFTView<x>:CURSor:HBArs:
 - DELTA?, 2-261
- DISplay:REFFFTView<x>:CURSor:MODe, 2-262
- DISplay:REFFFTView<x>:CURSor:
 - ONEOVERDELTAVALUE?, 2-262
- DISplay:REFFFTView<x>:CURSor:SCREEN:
 - AXPOsition, 2-263
- DISplay:REFFFTView<x>:CURSor:SCREEN:
 - AYPOsition, 2-263
- DISplay:REFFFTView<x>:CURSor:SCREEN:
 - BXPOsition, 2-264
- DISplay:REFFFTView<x>:CURSor:SCREEN:
 - BYPOsition, 2-264
- DISplay:REFFFTView<x>:CURSor:
 - SPLITMODE, 2-265
- DISplay:REFFFTView<x>:CURSor:STATE, 2-265
- DISplay:REFFFTView<x>:CURSor:VBArS:
 - APOsition, 2-266
- DISplay:REFFFTView<x>:CURSor:VBArS:
 - BPOsition, 2-266
- DISplay:REFFFTView<x>:CURSor:VBArS:
 - DELTA?, 2-267
- DISplay:REFFFTView<x>:CURSor:VBArS:
 - UNIts?, 2-267
- DISplay:REFFFTView<x>:CURSor:WAVEform:
 - AHPOsition?, 2-268

- DISplay:REFFFTView<x>:CURSor:WAVEform:
APOSition, 2-268
- DISplay:REFFFTView<x>:CURSor:WAVEform:
AVPOSition?, 2-268
- DISplay:REFFFTView<x>:CURSor:WAVEform:
BHPOSition?, 2-269
- DISplay:REFFFTView<x>:CURSor:WAVEform:
BPOSition, 2-269
- DISplay:REFFFTView<x>:CURSor:WAVEform:
BVPOSition?, 2-270
- DISplay:REFFFTView<x>:GRIDlines, 2-270
- DISplay:REFFFTView<x>:REF:REF<x>:
STATE, 2-271
- DISplay:REFFFTView<x>:XAXIS:SCALE, 2-271
- DISplay:REFFFTView<x>:ZOOM:XAXIS:
FROM, 2-272
- DISplay:REFFFTView<x>:ZOOM:XAXIS:TO, 2-272
- DISplay:REFFFTView<x>:ZOOM:YAXIS:
FROM, 2-273
- DISplay:REFFFTView<x>:ZOOM:YAXIS:TO, 2-273
- DISplay:SElect:BUS, 2-274
- DISplay:SElect:MATH, 2-274
- DISplay:SElect:REFERENCE, 2-275
- DISplay:SElect:SOURce, 2-275
- DISplay:SElect:VIEW, 2-276
- DISplay:SElect:WAVEView<x>:SOURce, 2-276
- DISplay:VARpersist, 2-276
- DISplay:WAVEform, 2-277
- DISplay:WAVEView:CURSor:CURSOR1:
ROLOCATION, 2-277
- DISplay:WAVEView:GRIDTYPE, 2-278
- DISplay:WAVEView<x>:BUS:B<x>:STATE, 2-278
- DISplay:WAVEView<x>:BUS:B<x>:VERTical:
POSition, 2-279
- DISplay:WAVEView<x>:CH<x>:STATE, 2-280
- DISplay:WAVEView<x>:CH<x>:VERTical:
POSition, 2-280
- DISplay:WAVEView<x>:CH<x>:VERTical:
SCALE, 2-281
- DISplay:WAVEView<x>:CURSor?, 2-281
- DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:AVPOSition?, 2-296
- DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:BVPOSition?, 2-297
- DISplay:WAVEView<x>:CURSor:
CURSOR<x>?, 2-282
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ASOURce, 2-283
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
BSOURce, 2-283
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
DDT?, 2-284
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
FUNction, 2-285
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArS:APOSition, 2-285
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArS:AUNIts?, 2-286
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArS:BPOSition, 2-286
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArS:BUNIts?, 2-287
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArS:DELTA?, 2-287
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
MODE, 2-288
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ONEOVERDELTAVALUE?, 2-289
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AXPOSition, 2-289
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AYPOSition, 2-290
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BXPOSition, 2-290
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BYPOSition, 2-291
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SPLITMODE, 2-292
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
STATE, 2-292
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBarS:APOSition, 2-293
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBarS:BPOSition, 2-293
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBarS:DELTA?, 2-294
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBarS:UNIts?, 2-294
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:APOSition, 2-295
- DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:BPOSition, 2-296

DISplay:WAVEView<x>:DCH<x>_D<x>:
 STATE, 2-297
 DISPLAY:WAVEVIEW<x>:DCH<x>_DALL:
 DIGORDER, 2-298
 DISplay:WAVEView<x>:DCH<x>_DALL:VERTical:
 POSition, 2-298
 DISplay:WAVEView<x>:FILTer, 2-299
 DISplay:WAVEView<x>:GRATicule, 2-299
 DISplay:WAVEView<x>:INTENSITY:
 GRATicule, 2-300
 DISplay:WAVEView<x>:INTENSITY:
 WAVEform, 2-300
 DISplay:WAVEView<x>:MATH:MATH<x>:
 AUTOScale, 2-301
 DISplay:WAVEView<x>:MATH:MATH<x>:
 STATE, 2-302
 DISplay:WAVEView<x>:MATH:MATH<x>:
 VERTical:POSition, 2-302
 DISplay:WAVEView<x>:MATH:MATH<x>:
 VERTical:SCALE, 2-303
 DISplay:WAVEView<x>:REF:REF<x>:
 STATE, 2-303
 DISplay:WAVEView<x>:REF:REF<x>:VERTical:
 POSition, 2-304
 DISplay:WAVEView<x>:REF:REF<x>:VERTical:
 SCALE, 2-304
 DISplay:WAVEView<x>:STYle, 2-305
 DISplay:WAVEView<x>:VIEWStYle, 2-305
 DISplay:WAVEView<x>:ZOOM?, 2-306
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>?, 2-306
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 HORizontal:POSition, 2-307
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 HORizontal:SCALE, 2-308
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 HORizontal:WINSCALE, 2-308
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 STATE, 2-309
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 VERTical:POSition, 2-310
 DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
 VERTical:SCALE, 2-310
 DISplay:WAVEView<y>:REF:REF<x>:
 FRAME, 2-311
 DISplay:WAVEView<y>:REF<x>_DALL:
 FRAME, 2-311

E

*ESE, 2-312
 *ESR?, 2-313
 ETHERnet:DHCPbootp, 2-313
 ETHERnet:DNS:IPADDRESS, 2-314
 ETHERnet:DOMAINname, 2-314
 ETHERnet:ENET:ADDRESS?, 2-315
 ETHERnet:GATEWay:IPADDRESS, 2-315
 ETHERnet:IPADDRESS, 2-315
 ETHERnet:LXI:LAN:RESET, 2-316
 ETHERnet:LXI:LAN:SERVICENAME, 2-316
 ETHERnet:LXI:LAN:STATus?, 2-317
 ETHERnet:NAME, 2-317
 ETHERnet:NETWORKCONFig, 2-318
 ETHERnet:PING, 2-318
 ETHERnet:PING:STATus?, 2-319
 ETHERnet:SUBNETMask, 2-319
 EVENT?, 2-320
 EVMsg?, 2-320
 EVQty?, 2-321

F

FACtory, 2-322
 FILESystem?, 2-322
 FILESystem:COpy, 2-323
 FILESystem:CWD, 2-324
 FILESystem:DELEte, 2-324
 FILESystem:DIR?, 2-325
 FILESystem:HOMEDir?, 2-326
 FILESystem:LDIR?, 2-326
 FILESystem:MKDir, 2-326
 FILESYSTEM:MOUNT:DRIVE, 2-327
 FILESystem:MOUNT:TEKDrive, 2-328
 FILESystem:READFile, 2-329
 FILESystem:REName, 2-329
 FILESystem:RMDir, 2-330
 FILESystem:TEKDrive:CODE?, 2-330
 FILESystem:TEKDrive:CODE:EXPIrytime?, 2-331
 FILESystem:TEKDrive:CODE:STATus?, 2-331
 FILESystem:UNMOUNT:DRIVE, 2-331
 FILESystem:UNMOUNT:TEKDrive, 2-332
 FILESystem:WRITEFile, 2-332
 FPAnel:PRESS, 2-333
 FPAnel:TURN, 2-333

H

HEADer, 2-334
 HORizontal?, 2-335
 HORizontal:ACQDURATION?, 2-335
 HORizontal:DELay:MODE, 2-335
 HORizontal:DELay:TIME, 2-336
 HORizontal:DIVisions?, 2-337
 HORizontal:MAIn:INTERPRatio?, 2-337
 HORizontal:MODE, 2-337
 HORizontal:MODE:MANual:CONFIgure, 2-338
 HORizontal:MODE:RECOrdlength, 2-338
 HORizontal:MODE:SAMPLERate, 2-339
 HORizontal:MODE:SCALE, 2-339
 HORizontal:POSition, 2-340
 HORizontal:PREViewstate?, 2-340
 HORizontal:RECOrdlength, 2-341
 HORizontal:SAMPLERate, 2-341
 HORizontal:SAMPLERate:ANALYZemode:
 MINimum:OVERRide, 2-341
 HORizontal:SAMPLERate:ANALYZemode:
 MINimum:VALue, 2-342
 HORizontal:SCALE, 2-343

I

ID?, 2-343
 *IDN?, 2-343

L

LIC:UNINSTALL?, 2-344
 LICense?, 2-344
 LICENSE:APPID?, 2-345
 LICense:COUNt?, 2-345
 LICense:ERRor?, 2-346
 LICense:GMT?, 2-346
 LICense:HID?, 2-346
 LICense:INSTall, 2-347
 LICense:ITEM?, 2-347
 LICense:LIST?, 2-348
 LICense:VALidate?, 2-348
 LOCK, 2-349
 *LRN?, 2-349

M

MAINWindow:BADGe:BRINGtoview, 2-350
 MAINWindow:FONTSize, 2-351

MAINWindow:RRBDisplaystate, 2-351
 MASK:DELete, 2-351
 MASK:MASK<x>:COUNT?, 2-352
 MASK:MASK<x>:COUNT:HITS?, 2-352
 MASK:MASK<x>:DEFinedby, 2-353
 MASK:MASK<x>:DISplay, 2-353
 MASK:MASK<x>:LIST?, 2-354
 MASK:MASK<x>:SEG<x>:POINTS, 2-355
 MASK:MASK<x>:SEG<x>COUNT:HITS?, 2-354
 MASK:MASK<x>:SOURce, 2-355
 MASK:MASK<x>:TEST:STATE, 2-356
 MASK:MASK<x>:TEST:STATUS?, 2-356
 MASK:MASK<x>:TEST:THReshold, 2-357
 MASK:MASK<x>:TOLerance:HORizontal, 2-357
 MASK:MASK<x>:TOLerance:UPDatenow, 2-358
 MASK:MASK<x>:TOLerance:VERTical, 2-358
 MASK:TEST:WAVEforms, 2-359
 MATH:ADDNew, 2-359
 MATH:DELete, 2-360
 MATH:LIST?, 2-360
 MATH:MATH<x>:AVG:MODE, 2-361
 MATH:MATH<x>:AVG:WEIGHT, 2-361
 MATH:MATH<x>:CAN:SUPPortedfields, 2-362
 MATH:MATH<x>:DEFine, 2-362
 MATH:MATH<x>:FUNCTion, 2-363
 MATH:MATH<x>:GATING, 2-364
 MATH:MATH<x>:I2C:SUPPortedfields, 2-363
 MATH:MATH<x>:INTERpolation, 2-364
 MATH:MATH<x>:LABel:COLor, 2-365
 MATH:MATH<x>:LABel:FONT:BOLD, 2-365
 MATH:MATH<x>:LABel:FONT:ITALic, 2-366
 MATH:MATH<x>:LABel:FONT:SIZE, 2-366
 MATH:MATH<x>:LABel:FONT:TYPE, 2-367
 MATH:MATH<x>:LABel:FONT:UNDERline, 2-367
 MATH:MATH<x>:LABel:NAME, 2-368
 MATH:MATH<x>:LABel:XPOS, 2-368
 MATH:MATH<x>:LABel:YPOS, 2-369
 MATH:MATH<x>:LIN:SUPPortedfields, 2-369
 MATH:MATH<x>:PARallel:SUPPortedfields, 2-370
 MATH:MATH<x>:RS232C:SUPPortedfields, 2-370
 MATH:MATH<x>:SENT:SUPPortedfields, 2-371
 MATH:MATH<x>:SIGNeddata, 2-371
 MATH:MATH<x>:SOURce<x>, 2-372
 MATH:MATH<x>:SPECTral:WINDow, 2-372
 MATH:MATH<x>:SPI:SUPPortedfields, 2-373
 MATH:MATH<x>:TYPE, 2-374
 MATH:MATH<x>:VUNIT, 2-375

MEASTABLE:ADDNew, 2-375
 MEASTABLE:DELETE, 2-375
 MEASUREMENT?, 2-376
 MEASUREMENT:ADDMEAS , 2-376
 MEASUREMENT:ADDNew, 2-379
 MEASUREMENT:ANNOTate, 2-380
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 FALLHigh, 2-380
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 FALLLow, 2-381
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 FALLMid, 2-381
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 HYSTEResis, 2-382
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 RISEHigh, 2-382
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 RISELow, 2-383
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 RISEMid, 2-383
 MEASUREMENT:CH<x>:REFLevels:ABSolute:
 TYPE, 2-384
 MEASUREMENT:CH<x>:REFLevels:BASETop, 2-384
 MEASUREMENT:CH<x>:REFLevels:METHod, 2-385
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 FALLHigh, 2-385
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 FALLLow, 2-386
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 FALLMid, 2-386
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 HYSTEResis, 2-387
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 RISEHigh, 2-387
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 RISELow, 2-388
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 RISEMid, 2-388
 MEASUREMENT:CH<x>:REFLevels:PERCent:
 TYPE, 2-389
 MEASUREMENT:DELEte , 2-390
 MEASUREMENT:DELETEALL , 2-390
 MEASUREMENT:EDGE<x>, 2-390
 MEASUREMENT:GATING, 2-391
 MEASUREMENT:GATING:ACTive, 2-391
 MEASUREMENT:GATING:ENDtime, 2-392
 MEASUREMENT:GATING:HYSTEResis, 2-392
 MEASUREMENT:GATING:LOGICSource, 2-393
 MEASUREMENT:GATING:MIDRef, 2-393
 MEASUREMENT:GATING:SEARCHSource, 2-394
 MEASUREMENT:GATING:STARTtime, 2-394
 MEASUREMENT:INTERp, 2-395
 MEASUREMENT:LIST?, 2-395
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 FALLHigh, 2-396
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 FALLLow, 2-396
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 FALLMid, 2-397
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 HYSTEResis, 2-397
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 RISEHigh, 2-398
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 RISELow, 2-398
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 RISEMid, 2-399
 MEASUREMENT:MATH<x>:REFLevels:ABSolute:
 TYPE, 2-399
 MEASUREMENT:MATH<x>:REFLevels:
 BASETop, 2-400
 MEASUREMENT:MATH<x>:REFLevels:
 METHod, 2-401
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 FALLHigh, 2-401
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 FALLLow, 2-402
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 FALLMid, 2-402
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 HYSTEResis, 2-403
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 RISEHigh, 2-404
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 RISELow, 2-404
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 RISEMid, 2-405
 MEASUREMENT:MATH<x>:REFLevels:PERCent:
 TYPE, 2-405
 MEASUREMENT:MEAS<x>:BURSTEDGTYPE, 2-406
 MEASUREMENT:MEAS<x>:CCRESULts:ALLAcqs:
 MAXimum?, 2-406
 MEASUREMENT:MEAS<x>:CCRESULts:ALLAcqs:
 MEAN?, 2-407

- MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:
MINimum?, 2-407
- MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:
PK2PK?, 2-408
- MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:
POPulation?, 2-408
- MEASUREMENT:MEAS<x>:CCRESULTS:ALLAcqs:
STDDev?, 2-409
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:MAXimum?, 2-409
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:MEAN?, 2-409
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:MINimum?, 2-410
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:PK2PK?, 2-410
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:POPulation?, 2-411
- MEASUREMENT:MEAS<x>:CCRESULTS:
CURRENTAcq:STDDev?, 2-411
- MEASUREMENT:MEAS<x>:DELAY:EDGE<x>, 2-411
- MEASUREMENT:MEAS<x>:DISPLAYstat:
ENABLE, 2-412
- MEASUREMENT:MEAS<x>:EDGE<x>, 2-416
- MEASUREMENT:MEAS<x>:EDGEIncr, 2-413
- MEASUREMENT:MEAS<x>:EDGES:
FROMLevel, 2-413
- MEASUREMENT:MEAS<x>:EDGES:LEVel, 2-414
- MEASUREMENT:MEAS<x>:EDGES:N, 2-414
- MEASUREMENT:MEAS<x>:EDGES:
SLEWRATEMethod, 2-415
- MEASUREMENT:MEAS<x>:EDGES:TOLevel, 2-415
- MEASUREMENT:MEAS<x>:FAILCount?, 2-416
- MEASUREMENT:MEAS<x>:FROMedge, 2-417
- MEASUREMENT:MEAS<x>:
FROMEDGESEARCHDIRect, 2-418
- MEASUREMENT:MEAS<x>:GATING, 2-418
- MEASUREMENT:MEAS<x>:GATING:ACTive, 2-419
- MEASUREMENT:MEAS<x>:GATING:ENDtime, 2-419
- MEASUREMENT:MEAS<x>:GATING:GLOBal, 2-420
- MEASUREMENT:MEAS<x>:GATING:
HYSTeresis, 2-421
- MEASUREMENT:MEAS<x>:GATING:
LOGICSource, 2-421
- MEASUREMENT:MEAS<x>:GATING:MIDRef, 2-422
- MEASUREMENT:MEAS<x>:GATING:
SEARCHSource, 2-422
- MEASUREMENT:MEAS<x>:GATING:
STARTtime, 2-423
- MEASUREMENT:MEAS<x>:GLOBalref, 2-423
- MEASUREMENT:MEAS<x>:HIGHREFVoltage, 2-424
- MEASUREMENT:MEAS<x>:IDLETime, 2-424
- MEASUREMENT:MEAS<x>:LABel, 2-425
- MEASUREMENT:MEAS<x>:LOWREFVoltage, 2-425
- MEASUREMENT:MEAS<x>:
PASSFAILENabled, 2-426
- MEASUREMENT:MEAS<x>:
PASSFAILHIGHlimit, 2-426
- MEASUREMENT:MEAS<x>:PASSFAILLIMIT, 2-427
- MEASUREMENT:MEAS<x>:
PASSFAILLOWlimit, 2-428
- MEASUREMENT:MEAS<x>:
PASSFAILMARGin, 2-429
- MEASUREMENT:MEAS<x>:PASSFAILWHEN, 2-429
- MEASUREMENT:MEAS<x>:PERFREQ:EDGE, 2-430
- MEASUREMENT:MEAS<x>:POLarity, 2-431
- MEASUREMENT:MEAS<x>:REFLevels:ABSolute:
FALLHigh, 2-431
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:FALLLow, 2-432
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:FALLMid, 2-433
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:HYSTeresis, 2-433
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:RISEHigh, 2-434
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:RISELow, 2-434
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:RISEMid, 2-435
- MEASUREMENT:MEAS<x>:REFLevels<x>:
ABSolute:TYPE, 2-435
- MEASUREMENT:MEAS<x>:REFLevels<x>:
BASETop, 2-436
- MEASUREMENT:MEAS<x>:REFLevels<x>:
METHod, 2-437
- MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:
FALLHigh, 2-437
- MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:
FALLLow, 2-438
- MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:
FALLMid, 2-438
- MEASUREMENT:MEAS<x>:REFLevels<x>:PERCent:
HYSTeresis, 2-439

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
 RISEHigh, 2-440
 MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
 RISELow, 2-440
 MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
 RISEMid, 2-441
 MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
 TYPE, 2-441
 MEASUrement:MEAS<x>:REFMode, 2-442
 MEASUrement:MEAS<x>:REFVoltage, 2-442
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 MAXimum?, 2-443
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 MEAN?, 2-443
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 MINimum?, 2-444
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 PK2PK?, 2-444
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 POPulation?, 2-445
 MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
 STDDev?, 2-445
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 MAXimum?, 2-445
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 MEAN?, 2-446
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 MINimum?, 2-446
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 PK2PK?, 2-447
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 POPulation?, 2-447
 MEASUrement:MEAS<x>:RESUlts:CURREntacq:
 STDDev?, 2-448
 MEASUrement:MEAS<x>:SIGNALType, 2-448
 MEASUrement:MEAS<x>:SOURce<x>, 2-449
 MEASUrement:MEAS<x>:STATUS?, 2-449
 MEASUrement:MEAS<x>:TOEdge, 2-450
 MEASUrement:MEAS<x>:
 TOEDGESEARCHDIRect, 2-450
 MEASUrement:MEAS<x>:TRANSition, 2-451
 MEASUrement:MEAS<x>:TYPE, 2-452
 MEASUrement:MEAS<x>:XUNIT?, 2-455
 MEASUrement:MEAS<x>:YUNIT?, 2-455
 MEASUrement:REF<x>:REFLevels:ABSolute:
 FALLHigh, 2-465
 MEASUrement:REF<x>:REFLevels:ABSolute:
 FALLLow, 2-466
 MEASUrement:REF<x>:REFLevels:ABSolute:
 FALLMid, 2-466
 MEASUrement:REF<x>:REFLevels:ABSolute:
 HYSteresis, 2-467
 MEASUrement:REF<x>:REFLevels:ABSolute:
 RISEHigh, 2-467
 MEASUrement:REF<x>:REFLevels:ABSolute:
 RISELow, 2-468
 MEASUrement:REF<x>:REFLevels:ABSolute:
 RISEMid, 2-468
 MEASUrement:REF<x>:REFLevels:ABSolute:
 TYPE, 2-469
 MEASUrement:REF<x>:REFLevels:
 BASETop, 2-469
 MEASUrement:REF<x>:REFLevels:METHod, 2-470
 MEASUrement:REF<x>:REFLevels:PERCent:
 FALLHigh, 2-471
 MEASUrement:REF<x>:REFLevels:PERCent:
 FALLLow, 2-471
 MEASUrement:REF<x>:REFLevels:PERCent:
 FALLMid, 2-472
 MEASUrement:REF<x>:REFLevels:PERCent:
 HYSteresis, 2-472
 MEASUrement:REF<x>:REFLevels:PERCent:
 RISEHigh, 2-473
 MEASUrement:REF<x>:REFLevels:PERCent:
 RISELow, 2-473
 MEASUrement:REF<x>:REFLevels:PERCent:
 RISEMid, 2-474
 MEASUrement:REF<x>:REFLevels:PERCent:
 TYPE, 2-474
 MEASUrement:REFLevels:ABSolute:
 FALLHigh, 2-455
 MEASUrement:REFLevels:ABSolute:
 FALLLow, 2-456
 MEASUrement:REFLevels:ABSolute:
 FALLMid, 2-456
 MEASUrement:REFLevels:ABSolute:
 HYSteresis, 2-457
 MEASUrement:REFLevels:ABSolute:
 RISEHigh, 2-457
 MEASUrement:REFLevels:ABSolute:
 RISELow, 2-458
 MEASUrement:REFLevels:ABSolute:
 RISEMid, 2-458

MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE, 2-459
 MEASUREMENT:REFLEVELS:BASETOP, 2-459
 MEASUREMENT:REFLEVELS:METHOd, 2-460
 MEASUREMENT:REFLEVELS:MODE, 2-460
 MEASUREMENT:REFLEVELS:PERCENT:
 FALLHigh, 2-461
 MEASUREMENT:REFLEVELS:PERCENT:
 FALLLow, 2-461
 MEASUREMENT:REFLEVELS:PERCENT:
 FALLMid, 2-462
 MEASUREMENT:REFLEVELS:PERCENT:
 HYSTERESIS, 2-462
 MEASUREMENT:REFLEVELS:PERCENT:
 RISEHigh, 2-463
 MEASUREMENT:REFLEVELS:PERCENT:
 RISELow, 2-463
 MEASUREMENT:REFLEVELS:PERCENT:
 RISEMid, 2-464
 MEASUREMENT:REFLEVELS:PERCENT:TYPE, 2-464
 MEASUREMENT:REFLEVELS:TYPE, 2-465
 MEASUREMENT:STATISTICS:CYCLEMODE, 2-475

N

NEWpass, 2-475

O

*OPC, 2-476
 *OPT?, 2-476

P

PASSWord, 2-477
 PAUSE, 2-478
 PG:AMPLitude, 2-478
 PG:BIT:ONE, 2-479
 PG:BIT:THREE, 2-479
 PG:BIT:TWO, 2-480
 PG:BIT:ZERO, 2-480
 PG:BITRate, 2-481
 PG:BURSt:CCOUnt, 2-481
 PG:BURSt:TRIGger, 2-482
 PG:FILE:PATtern, 2-482
 PG:OUTPut:MODE, 2-483
 PG:PATTERNdefinition, 2-484
 PLOT:ADDNew, 2-484
 PLOT:DELete, 2-484
 PLOT:LIST?, 2-485

PLOT:PLOT<x>:SOURce<x>, 2-485
 PLOT:PLOT<x>:TYPE, 2-486
 *PSC, 2-486
 *PUD, 2-487

R

RECALL:MASK, 2-487
 RECALL:SESSion, 2-488
 RECALL:SETUp, 2-489
 RECALL:WAVEform, 2-490
 REF:ADDNew, 2-490
 REF:DELete, 2-491
 REF:LIST?, 2-491
 REF:REF<x>:DESKew, 2-491
 REF:REF<x>:LAbel:COLor, 2-492
 REF:REF<x>:LAbel:FONT:BOLD, 2-492
 REF:REF<x>:LAbel:FONT:ITALic, 2-493
 REF:REF<x>:LAbel:FONT:SIZE, 2-493
 REF:REF<x>:LAbel:FONT:TYPE, 2-494
 REF:REF<x>:LAbel:FONT:UNDERline, 2-494
 REF:REF<x>:LAbel:NAME, 2-494
 REF:REF<x>:LAbel:XPOS, 2-495
 REF:REF<x>:LAbel:YPOS, 2-495
 REF:REF<x>:SOURce, 2-496
 REM, 2-496
 ROSc:SOURce, 2-497
 ROSc:STATE?, 2-497
 *RST, 2-498

S

SAVe:EVENTtable:BUS, 2-500
 SAVe:EVENTtable:MEASUREMENT, 2-500
 SAVe:EVENTtable:SEARCHTable, 2-501
 SAVe:IMAGe, 2-501
 SAVe:IMAGe:COMPosition, 2-502
 SAVe:IMAGe:VIEWTYpe, 2-502
 SAVe:PLOTData, 2-503
 SAVe:REPOrt, 2-504
 SAVe:REPOrt:COMMENTS, 2-504
 SAVe:SESSion, 2-505
 SAVe:SETUp, 2-505
 SAVe:SETUp:INCLUDEREFs, 2-506
 SAVe:WAVEform, 2-507
 SAVe:WAVEform:GATing, 2-508
 SAVe:WAVEform:GATing:RESAMPLERate, 2-509
 SAVe:WAVEform:SOURCELst?, 2-510

- SAVEON:FILE:DEST, 2-510
 SAVEON:FILE:NAME, 2-511
 SAVEON:IMAGe, 2-511
 SAVEON:IMAGe:FILEFormat, 2-512
 SAVEON:TRIGger, 2-512
 SAVEON:WAVEform, 2-513
 SAVEON:WAVEform:FILEFormat, 2-514
 SAVEON:WAVEform:SOURce, 2-515
 SAVEONEVent:FILEDest, 2-515
 SAVEONEVent:FILEName, 2-516
 SAVEONEVent:IMAGe:FILEFormat, 2-516
 SAVEONEVent:WAVEform:FILEFormat, 2-517
 SAVEONEVent:WAVEform:SOURce, 2-518
 SCOPEApp REBOOT, 2-518
 SEARCH:ADDNew, 2-519
 SEARCH:DELeTe, 2-519
 SEARCH:DELETEALL, 2-519
 SEARCH:LIST?, 2-520
 SEARCH:SEARCH<x>:COpy, 2-520
 SEARCH:SEARCH<x>:NAVigate, 2-520
 SEARCH:SEARCH<x>:TOTAL?, 2-521
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 CONDition, 2-521
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 DATa:DIRectioN, 2-522
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 DATa:OFFSet, 2-522
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 DATa:QUALifier, 2-523
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 DATa:SIZE, 2-524
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 DATa:VALue, 2-524
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 ERRType, 2-525
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:
 BRSBit, 2-526
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:
 ESIBit, 2-526
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 FRAMeType, 2-527
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 IDentifier:MODE, 2-528
 SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
 IDentifier:VALue, 2-528
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 ADDRess:MODE, 2-529
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 ADDRess:VALue, 2-529
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 CONDition, 2-530
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 DATA:DIRectioN, 2-531
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 DATA:SIZE, 2-531
 SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:
 DATA:VALue, 2-532
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 CONDition, 2-533
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 DATA:HIVALue, 2-533
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 DATA:QUALifier, 2-534
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 DATA:SIZE, 2-534
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 DATA:VALue, 2-535
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 ERRTYPE, 2-535
 SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:
 IDentifier:VALue, 2-536
 SEARCH:SEARCH<x>:TRIGger:A:BUS:PARAllel:
 DATA:VALue, 2-536
 SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:
 CONDition, 2-537
 SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:
 DATA:SIZE, 2-538
 SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:
 DATA:VALue, 2-538
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 CONDition, 2-539
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 ERRType, 2-539
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 ERRType:CRC, 2-540
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 FAST:CHAN1A:HIVALue, 2-541
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 FAST:CHAN1A:QUALifier, 2-542
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 FAST:CHAN1A:VALue, 2-543
 SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
 FAST:CHAN2B:HIVALue, 2-543

- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:CHAN2B:QUALifier, 2-544
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:CHAN2B:VALue, 2-545
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:COUNTer:HIVALue, 2-546
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:COUNTer:QUALifier, 2-547
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:COUNTer:VALue, 2-548
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:INVERTNIBble:VALue, 2-548
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
FAST:STATus:VALue, 2-549
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
PAUSE:QUALifier, 2-550
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
PAUSE:TICKs:HIVALue, 2-551
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
PAUSE:TICKs:VALue, 2-551
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
SLOW:DATA:HIVALue, 2-552
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
SLOW:DATA:QUALifier, 2-553
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
SLOW:DATA:VALue, 2-554
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:
SLOW:IDentifier:VALue, 2-554
- SEARCH:SEARCH<x>:TRIGger:A:BUS:
SOUrce, 2-555
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:
CONDition, 2-556
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:
SIZE, 2-556
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATA:
VALue, 2-557
- SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:
SOURCEType, 2-557
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:
SLOpe, 2-558
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:
SOUrce, 2-559
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:
THReshold, 2-559
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCK:
THReshold, 2-560
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
DELTatime, 2-560
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
FUNCTion, 2-561
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:
CLOCK:SOUrce, 2-561
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:
CH<x>, 2-562
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:
MATH<x>, 2-562
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:
REF<x>, 2-563
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
LOGICPattern:CH<x>, 2-563
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
LOGICPattern:DCH<x>_D<x>, 2-564
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
LOGICPattern:MATH<x>, 2-565
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
LOGICPattern:REF<x>, 2-565
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
POLarity, 2-566
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
USEClockedge, 2-566
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:
WHEn, 2-567
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
HIGHLimit, 2-568
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
LOGICQUALification, 2-568
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
LOWLimit, 2-569
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
POLarity, 2-569
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
SOUrce, 2-570
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
THReshold, 2-570
- SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:
WHEn, 2-571
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
LOGICQUALification, 2-572
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
POLarity, 2-572
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
SOUrce, 2-573

- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
THReshold:HIGH, 2-573
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
THReshold:LOW, 2-574
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
WHEn, 2-574
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:
WIDth, 2-575
- SEARCH:SEARCH<x>:TRIGger:A:SETHold
:LOGICPattern:DCH<x>_D<x>, 2-580
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
CLOCK:EDGE, 2-575
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
CLOCK:SOUrce, 2-576
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
CLOCK:THReshold, 2-577
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
HOLDTime, 2-577
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LEVel:CH<x>, 2-578
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LEVel:MATH<x>, 2-578
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LEVel:REF<x>, 2-579
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LOGICPattern:CH<x>, 2-579
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LOGICPattern:MATH<x>, 2-580
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
LOGICPattern:REF<x>, 2-581
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:
SETTime, 2-582
- SEARCH:SEARCH<x>:TRIGger:A:STATE, 2-582
- SEARCH:SEARCH<x>:TRIGger:A:STOPAcq, 2-583
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
LOGICQUALification, 2-583
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
POLarity, 2-584
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
SOUrce, 2-584
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
THReshold, 2-585
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
TIME, 2-585
- SEARCH:SEARCH<x>:TRIGger:A:TYPE, 2-586
- SEARCH:SELected, 2-587
- SELect:DCH<x>:DAll, 2-587
- SET?, 2-588
- SOCKETServer:ENable, 2-589
- SOCKETServer:PORT, 2-589
- SOCKETServer:PROTOCOL, 2-590
- *SRE, 2-591
- *STB?, 2-591
- ## T
- TEKSecure, 2-592
- TIME, 2-592
- TIME:ZONE, 2-593
- TIME:ZONE:UTCDELTA, 2-593
- TOTALuptime?, 2-594
- TOUCHSCreen:STATe, 2-594
- *TRG, 2-595
- TRIGger, 2-595
- TRIGger:A, 2-596
- TRIGger:A:BUS:B<x>:CAN:CONDition, 2-596
- TRIGger:A:BUS:B<x>:CAN:DATA:DIRection, 2-597
- TRIGger:A:BUS:B<x>:CAN:DATA:OFFSet, 2-598
- TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier, 2-598
- TRIGger:A:BUS:B<x>:CAN:DATA:SIZE, 2-599
- TRIGger:A:BUS:B<x>:CAN:DATA:VALue, 2-600
- TRIGger:A:BUS:B<x>:CAN:ERRType, 2-601
- TRIGger:A:BUS:B<x>:CAN:FD:BRSBit, 2-601
- TRIGger:A:BUS:B<x>:CAN:FD:ESIBit, 2-602
- TRIGger:A:BUS:B<x>:CAN:FRAMeType, 2-603
- TRIGger:A:BUS:B<x>:CAN:IDentifier:MODE, 2-603
- TRIGger:A:BUS:B<x>:CAN:IDentifier:
VALue, 2-604
- TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE, 2-604
- TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue, 2-605
- TRIGger:A:BUS:B<x>:I2C:CONDition, 2-606
- TRIGger:A:BUS:B<x>:I2C:DATA:DIRection, 2-606
- TRIGger:A:BUS:B<x>:I2C:DATA:SIZE, 2-607
- TRIGger:A:BUS:B<x>:I2C:DATA:VALue, 2-607
- TRIGger:A:BUS:B<x>:LIN:CONDition, 2-608
- TRIGger:A:BUS:B<x>:LIN:DATA:HIVALue, 2-609
- TRIGger:A:BUS:B<x>:LIN:DATA:QUALifier, 2-609
- TRIGger:A:BUS:B<x>:LIN:DATA:SIZE, 2-610
- TRIGger:A:BUS:B<x>:LIN:DATA:VALue, 2-611
- TRIGger:A:BUS:B<x>:LIN:ERRType, 2-611
- TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue, 2-612
- TRIGger:A:BUS:B<x>:PARallel:DATA:VALue, 2-612
- TRIGger:A:BUS:B<x>:RS232C:CONDition, 2-613
- TRIGger:A:BUS:B<x>:RS232C:DATA:SIZE, 2-614
- TRIGger:A:BUS:B<x>:RS232C:DATA:VALue, 2-614

- TRIGger:A:BUS:B<x>:SENT:CONDition, 2-615
 TRIGger:A:BUS:B<x>:SENT:ERRType, 2-615
 TRIGger:A:BUS:B<x>:SENT:ERRType:CRC, 2-616
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
 HIVALue, 2-617
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
 QUALifier, 2-617
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
 VALue, 2-618
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
 HIVALue, 2-619
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
 QUALifier, 2-620
 TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
 VALue, 2-621
 TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:
 HIVALue, 2-621
 TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:
 QUALifier, 2-622
 TRIGger:A:BUS:B<x>:SENT:FAST:COUNTER:
 VALue, 2-623
 TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:
 VALue, 2-624
 TRIGger:A:BUS:B<x>:SENT:FAST:STATus:
 VALue, 2-625
 TRIGger:A:BUS:B<x>:SENT:PAUSE:
 QUALifier, 2-625
 TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
 HIVALue, 2-626
 TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
 QUALifier, 2-627
 TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
 VALue, 2-628
 TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:
 VALue, 2-629
 TRIGger:A:BUS:B<x>:SPI:CONDition, 2-629
 TRIGger:A:BUS:B<x>:SPI:DATA:SIZE, 2-630
 TRIGger:A:BUS:B<x>:SPI:DATA:VALue, 2-630
 TRIGger:A:BUS:SOURce, 2-631
 TRIGger:A:EDGE:COUPling, 2-631
 TRIGger:A:EDGE:SLOpe, 2-632
 TRIGger:A:EDGE:SOURce, 2-633
 TRIGger:A:HOLDoff:TIME, 2-651
 TRIGger:A:LEVel:CH<x>, 2-633
 TRIGger:A:LOGIc:DELTAtime, 2-634
 TRIGger:A:LOGIc:FUNCTion, 2-634
 TRIGger:A:LOGIc:INPut:CLOCK:SOURce, 2-635
 TRIGger:A:LOGIc:POLarity, 2-636
 TRIGger:A:LOGIc:USECLocked, 2-636
 TRIGger:A:LOGIc:WHEN, 2-637
 TRIGger:A:LOGICPattern:CH<x>, 2-637
 TRIGger:A:LOGICPattern:DCH<x>_D<x>, 2-638
 TRIGger:A:LOWerthreshold:CH<x>, 2-639
 TRIGger:A:MODE, 2-651
 TRIGger:A:PULSEWidth:HIGHLimit, 2-639
 TRIGger:A:PULSEWidth:LOWLimit, 2-640
 TRIGger:A:PULSEWidth:POLarity, 2-640
 TRIGger:A:PULSEWidth:SOURce, 2-641
 TRIGger:A:PULSEWidth:WHEN, 2-641
 TRIGger:A:RUNT:POLarity, 2-642
 TRIGger:A:RUNT:SOURce, 2-643
 TRIGger:A:RUNT:WHEN, 2-644
 TRIGger:A:RUNT:WIDTH, 2-644
 TRIGger:A:SETHold:CLOCK:EDGE, 2-645
 TRIGger:A:SETHold:CLOCK:SOURce, 2-645
 TRIGger:A:SETHold:HOLDTime, 2-646
 TRIGger:A:SETHold:SETTime, 2-646
 TRIGGER:A:SETHOLDLOGICVAL:
 DCH<x>_D<x>, 2-647
 TRIGger:A:TIMEOut:POLarity, 2-648
 TRIGger:A:TIMEOut:SOURce, 2-648
 TRIGger:A:TIMEOut:TIME, 2-649
 TRIGger:A:TYPE, 2-649
 TRIGger:A:UPPerthreshold:CH<x>, 2-650
 TRIGger:AUXLevel, 2-652
 TRIGger:HYSteresis:USER:STATe, 2-652
 TRIGger:HYSteresis:USER:VALue, 2-653
 TRIGger:STATE?, 2-653
 *TST?, 2-654
- ## U
- UNDO, 2-654
 UNLock, 2-655
 USBDevice:CONFigure, 2-655
- ## V
- VERBose, 2-656
 VERTical:DESKew:FROM:
 CUSTOMPROPagation, 2-657
 VERTical:DESKew:FROM:SOURce, 2-657
 VERTical:DESKew:STATIC, 2-658
 VERTical:DESKew:TO:
 CUSTOMPROPagation, 2-658

VERTical:DESKEW:TO:SOUrce, 2-659

W

*WAI, 2-659

WAVFrm?, 2-660

WFMOutpre?, 2-660

WFMOutpre:ASC_Fmt?, 2-661

WFMOutpre:BIT_Nr, 2-661

WFMOutpre:BN_Fmt, 2-662

WFMOutpre:BYT_Nr, 2-663

WFMOutpre:BYT_Or, 2-663

WFMOutpre:CENTERFREQuency?, 2-664

WFMOutpre:DOMain?, 2-664

WFMOutpre:ENCdg, 2-665

WFMOutpre:NR_Pt?, 2-666

WFMOutpre:PT_Fmt?, 2-666

WFMOutpre:PT_Off?, 2-667

WFMOutpre:PT_ORDER?, 2-667

WFMOutpre:RESample, 2-668

WFMOutpre:SPAN?, 2-668

WFMOutpre:WFIId?, 2-669

WFMOutpre:WFMTYPE?, 2-670

WFMOutpre:XINcr?, 2-671

WFMOutpre:XUNit?, 2-671

WFMOutpre:XZEro?, 2-672

WFMOutpre:YMUlt?, 2-672

WFMOutpre:YOff?, 2-673

WFMOutpre:YUNit?, 2-673

WFMOutpre:YZEro?, 2-674