Tektronix®

2 Series Mixed Signal Oscilloscope MSO22 and MSO24

Programmer Manual

077-1776-07

Tektronix®

2 Series Mixed Signal Oscilloscope MSO22 and MSO24

Programmer Manual

Supports FW version 2.8 and above

www.tek.com

077-1776-07

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 June 2023.

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 2-12
AFG Command Group 2-13
Alias command group 2-14
Battery command group 2-15
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 2-38
Miscellaneous command group 2-48
Pattern Generator group 2-51

Plot command group 2-52
Power command group..... 2-52
Save and Recall command group 2-54
Save On command Group 2-55
Search and Mark command group 2-56
Self Test command group..... 2-62
Status and Error command group 2-63
Trigger command group 2-64
Vertical command group 2-68
Waveform Transfer command group 2-70
Zoom command group 2-74
Commands listed in alphabetical order..... 2-75
Status and Events 3-1
Registers 3-1
Queues 3-4
Event Handling Sequence..... 3-5
Synchronization Methods..... 3-7
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 " `Telnet` " 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 `*idn?`

    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:ADDRess: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:ONEOVERDELTATVALUE	:DISplay:WAVEView<n>:CURSor:CURSOR:ONEOVERDELTATVALUE
: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|:TRIGAfter)	: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:RECOrdlengt
:HORizontal:RESOlution	
:HORizontal:DIGital:RECOrdlength	
:HORizontal:DIGital:RECOrdlength:MAIn	
:HORizontal:ACQLENGTH"	
:CH<X>:AMPSViavolts:ENABle	:CH<X>:PROBEFunc:EXTUnits:STATE
:CH<X>:AMPSViavolts: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.

Some queries cause the instrument to return results with 9.91E+37. This result represents a NaN (Not a number) and may indicate an error in performing an instrument query operation.

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.

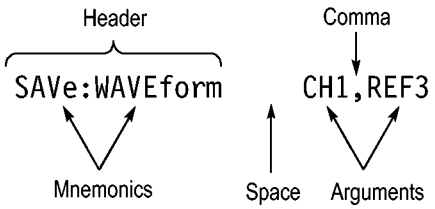
Headers

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

Command Parts



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:FASTAcq:PALEtte	Sets or queries the waveform grading for fast acquisition mode.
ACQuire:FASTAcq:STATE	Sets or queries the state of fast acquisition mode.
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 `ACQuire:STATE RUN` 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.

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

Command	Description
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.
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.

Table 2-19: AFG commands (cont.)

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

Table 2-23: Bus commands (cont.)

Command	Description
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.
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<x>SOUrce:THReshold	Sets or queries the specified bit source threshold for the specified parallel bus.

Table 2-23: Bus commands (cont.)

Command	Description
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.
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:NUMCHANnel	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.

Table 2-23: Bus commands (cont.)

Command	Description
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.
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:POSition	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:FACtory:STATus?	Returns the factory calibration status value saved in nonvolatile memory.
CALibrate:INTERNal	Starts the signal path calibration. Takes several minutes to run.
CALibrate:INTERNal:DATE?	Returns the date and time of the most recent signal path calibration.
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
CALLOUTS:ADDNew	Creates a new note.
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE	Sets or queries the source of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS	Sets or queries the X-Position of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:COLOR	Sets or queries the text color of the callout.
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:X	Sets or queries horizontal display position of the callout text
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:Y	Sets or queries vertical display position of the callout text
CALLOUTS:CALLOUT<x>:FONT:BOLD	Sets or queries the bold state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:ITALIC	Sets or queries the italic state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:SIZE	Sets or queries the font size of the callout text.
CALLOUTS:CALLOUT<x>:FONT:TYPE	Sets or queries the font type of the callout.
CALLOUTS:CALLOUT<x>:FONT:UNDERLine	Sets or queries the underline state of the callout text.
CALLOUTS:CALLOUT<x>:TEXT	Sets or queries the callout text.
CALLOUTS:CALLOUT<x>:TYPE	Sets or queries type of the callout
CALLOUTS:DELete	Deletes the specified 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
DISplay:MATHFFTView<x>:CURSor:ASOUrce?	Queries the cursor source for cursor A in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:BSOUrce?	Queries the cursor source for cursor B in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:FUNCtion	Sets or queries the cursor typein the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition	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:ONEOVERDELTATVALUE?	Queries the one over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>: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 positionin 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:ONEOVERDELTATVALUE?	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>: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:ONEOVERDELTATVALUE?	Queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>: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>:ONEOVERDELTATVALUE?	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>:DAll	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:POSition	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:POSition	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:POSition	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: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.
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:POSition	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
MASK:DELete	Deletes all mask segments of the specified mask test.
MASK:MASK<x>:COUNT?	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.
MASK:MASK<x>:COUNT:HITS?	Returns the total number of mask hits in all mask segments for the specified mask test.
MASK:MASK<x>:DEFinedby	Sets or queries whether the specified mask is defined by segments or tolerances.
MASK:MASK<x>:DISplay	Sets or queries the display state of the mask used for the specified mask test.
MASK:MASK<x>:LIST?	Queries the list of segments in the mask used by the specified mask test.
MASK:MASK<x>:SEG<x>:POINTS	Sets or queries the X/Y coordinates of all points in the designated mask segment.
MASK:MASK<x>:COUNT:SEG<x>:HITS?	Returns the total number of mask hits in the specified mask segment of the specified mask test.
MASK:MASK<x>:SOUrce	Sets or queries analog source for the specified mask test.
MASK:MASK<x>:TESt:CTHReshold	Sets or queries the number of consecutive waveform violations needed for the specified mask test to change from PASS to FAIL.
MASK:MASK<x>:TESt:STATE	Sets or queries state of the specified mask test.
MASK:MASK<x>:TESt:STATUS?	Queries the status of the specified mask test.
MASK:MASK<x>:TESt:THReshold	Sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.
MASK:MASK<x>:TOLerance:HABSolute	Sets or queries the mask horizontal tolerance in absolute units of seconds.
MASK:MASK<x>:TOLerance:HORizontal	Sets or queries the mask horizontal tolerance.
MASK:MASK<x>:TOLerance:UPDatenow	Causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.
MASK:MASK<x>:TOLerance:VABSolute	Sets or queries the mask horizontal tolerance in absolute units of seconds. This value is used when the mask TTYPe is set to ABSolute and the UPDatenow command is sent.
MASK:MASK<x>:TOLerance:VERTical	Sets or queries the mask vertical tolerance.
MASK:MASK<x>:TTYPe	Sets or queries the type of tolerance values used when defining a tolerance mask.
MASK:TESt:WAVEforms	Sets or queries the number of waveform acquisitions to test during mask testing.
RECAll:MASK	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:HORZ	Sets or queries the horizontal display scale of the spectral math waveform.
MATH:MATH<x>:SPECTral:MAG	Sets or queries the units of the SpectralMag function in the specified math definition string.
MATH:MATH<x>:SPECTral:PHASE	Sets or queries the units of a SpectralPhase function in the specified math definition string.
MATH:MATH<x>:SPECTral:SOUrce	Sets or queries the specified spectral math source.
MATH:MATH<x>:SPECTral:SUPPress	Sets or queries whether suppression threshold for the specified math waveform is enabled.
MATH:MATH<x>:SPECTral:SUPPress:VALue	Sets or queries in volts the value of suppression threshold of the specified math waveform.
MATH:MATH<x>:SPECTral:TYPE	Sets or queries the FFT type selected for spectral analysis.
MATH:MATH<x>:SPECTral:UNWRap	Sets or queries whether phase unwrap of the spectral analyzer output data is enabled.
MATH:MATH<x>:SPECTral:UNWRap:DEGrees	Sets or queries how many degrees adjacent phase values can jump before being unwrapped.
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.

Table 2-33: Math commands (cont.)

Command	Description
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
CUSTOMTABle:ADDNew	This command adds new custom results table.
CUSTOMTABle:DELete	This command deletes the custom result(s) table that was added.
CUSTOMTABle:LIST?	This command queries the list of custom result tables has been added.
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.
MEASTABle:LIST?	Returns all measurement results view tables.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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: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.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:CCRESULts:ALLAcqs:MAXimum?	This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions.
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>:CYCLemode	Sets or queries the cycle mode for the measurement.
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>:EDGEIncre	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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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>:TOEdge	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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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.

Table 2-34: Measurement commands (cont.)

Command	Description
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.
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.
SECurity:LOGger:STATe	Sets or queries the state of the security log.
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.

Table 2-35: Miscellaneous commands (cont.)

Command	Description
TOUCHSCReen:STATe	Sets or queries the enabled state of the touch screen.
*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.
VXI:ENAble	Sets or queries the state of the VXI-11 server.
VXI:PORT:HIGH	Sets or queries the higher end of the port range for the VXI-11 server.
VXI:PORT:LOW	Sets or queries the lower end of the port range for the VXI-11 server.

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.

Power command group

Table 2-38: Power commands

Command	Description
POWer:ADDNew	Adds the specified power measurement badge.
POWer:DELete	Deletes the specified power measurement badge.
POWer:POWer<x>:CLRESPONSE:AMP<x>Val	Sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:AMPMode	Sets or queries the amplitude mode for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:ANALYSISMethod?	Sets or queries the Analysis Method for Control Loop Response measurement.
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?	Queries connection status to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude	Sets or queries the constant amplitude voltage for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val	Sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENerator	Sets or queries the generator source for the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENIPADDress	Sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.
POWer:POWer<x>:CLRESPONSE:IMPEDance	Sets or queries the vertical termination impedance for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:INPUTSOurce	Sets or queries the input source for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce	Sets or queries the output source for the Control Loop Response power measurement.

Table 2-38: Power commands (cont.)

Command	Description
POWer:POWer<x>:CLRESPONSE:PPD	Sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:STARTFREQuency	Sets or queries the start frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:STOPFREQuency	Sets or queries the stop frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:TESTCONNection	Tests the connection to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:PRESET	Sets or queries the input source for positive pulse width measurement of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum?	Queries the maximum value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MEAN?	Queries the mean value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MINimum?	Queries the minimum value of the current acquisition for the measurement parameter of the specified power measurement number.

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 2-39: Save and Recall commands

Command	Description
FACtory	Resets the instrument to factory default settings.
RECAll:SESsion	Restores the state of the instrument from a saved session file.
RECAll:SETUp	Recalls saved instrument settings.
RECAll: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:MASK	Saves the given waveview mask to the specified file.
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 2-40: 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:TRIGer 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 2-41: 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 2-41: 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 2-41: 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 2-41: 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:DELTatime	Specifies the Logic search delta time value.

Table 2-41: 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.
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.

Table 2-41: Search and Mark commands (cont.)

Command	Description
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 2-42: 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 2-43: 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 2-44: 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 I²C address mode to 7 or 10-bit.
TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue	Specifies the binary address string used for the I²C trigger if the trigger condition is ADDRESS or ADDRANDDATA.
TRIGger:A:BUS:B<x>:I2C:CONDition	Specifies the trigger condition for an I²C trigger.
TRIGger:A:BUS:B<x>:I2C:DATa:DIRection	Specifies the I²C 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 I2C trigger if the trigger condition is DATA or ADDRANDDATA.

Table 2-44: 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 OUTRANGE.
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 2-44: 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 2-44: 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 `TIMEOut` trigger type, this commands specifies the polarity to be used.
TRIGger:A:TIMEOut:SOUrce	When triggering using the `TIMEOut` trigger type, this command specifies the source.
TRIGger:A:TIMEOut:TIMe	When triggering using the `TIMEOut` trigger type, this command specifies the timeout time, in seconds.
TRIGger:A:TRANsition:DELTatime	Specifies the delta time (that is Time Limit) used in calculating the transition value for the transition (that is Rise or Fall Time) trigger.
TRIGger:A:TRANsition:LOGICQUALification	Specifies whether or not to use logic qualification for a transition trigger.
TRIGger:A:TRANsition:POLarity	Specifies the polarity for the transition trigger.
TRIGger:A:TRANsition:SOUrce	Specifies the source waveform for a transition trigger.
TRIGger:A:TRANsition:WHEn	Specifies whether to check for a transitioning signal that is faster or slower than the specified delta time.
TRIGger:A:TYPe	Sets or queries the type of A trigger.
TRIGger:A:UPPerthreshold: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 2-45: 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 2-45: 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 ≥ 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.
SELect:CH<x>	Turns on or off the specified waveform or returns whether the specified channel is on or off
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 WFMOutpre? 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 `WFMOutpre: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 `WFMOutpre`.
6. Transfer waveform data from the instrument using `CURVe?`.

Table 2-46: Waveform Transfer commands

Command	Description
CURVe?	The query format transfers waveform data from instrument specified by the `DATa:SOUrce` command.
CURVEStream?	Continuously transfers waveform data from the instrument as it is acquired.
DATa	Sets the format and location of the waveform data that is transferred with the `CURVe` Command. Or queries the format and location of the waveform data that is transferred with the `CURVe?` command.
DATa:ENCdg	Sets or queries the format of outgoing waveform data.
DATa:MODe	Sets or returns the mode for waveform data.
DATa:RESample	Sets or queries the resampling of outgoing waveform data.
DATa:SOUrce	Sets or queries the location of waveform data transferred from the instrument.
DATa:SOUrce:AVAILable?	Returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries.
DATa:STARt	Sets or queries the starting point in waveform transfer.
DATa:STOP	Sets or queries the ending data point in waveform transfer.
DATa:WIDth	Specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVe? query.
WAVFrm?	Returns `WFMOutpre?` and `CURVe?` data for the waveform as specified by the `DATA:SOUrce` command.
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command.
WFMOutpre:ASC_Fmt?	Returns the format for ASCII data transferred from the instrument.

Table 2-46: Waveform Transfer commands (cont.)

Command	Description
WFMOutpre:BIT_Nr	Sets or queries the number of bits per waveform point that outgoing waveforms contain.
WFMOutpre:BN_Fmt	Sets or queries the format of binary data for the waveform.
WFMOutpre:BYT_Nr	Sets or queries the data width for the waveform.
WFMOutpre:BYT_Or	Sets or queries the byte order of waveform points.
WFMOutpre:CENTERFREQuency?	Returns the center frequency of the outgoing waveform.
WFMOutpre:DOMain?	Returns the domain of the outgoing waveform.
WFMOutpre:ENCdg	Sets or queries the type of encoding for outgoing waveforms.
WFMOutpre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query.
WFMOutpre:PT_Fmt?	Returns the point format for the waveform.
WFMOutpre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform.
WFMOutpre:PT_ORder?	Returns whether the source waveform is Fast Acquisition.
WFMOutpre:RESample	Sets or queries the resampling of outgoing waveform data.
WFMOutpre:SPAN?	Returns the frequency span of the outgoing waveform.
WFMOutpre:WFId?	Returns a string describing the acquisition parameters for the waveform.
WFMOutpre:WFMTYPe?	Returns the type of the outgoing waveform.
WFMOutpre:XINcr?	Returns the horizontal sampling interval.
WFMOutpre:XUNit?	Returns the horizontal units for the waveform.
WFMOutpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger.
WFMOutpre:YOFf?	Returns the vertical offset in digitizing levels for the waveform.
WFMOutpre:YUNit?	Returns the vertical units for the waveform.
WFMOutpre:YZEro?	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 2-47: 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:FASTAcq:PALEtte

Sets or queries the waveform grading for fast acquisition mode.

Group Acquisition

Syntax
```
ACQuire:FASTAcq:PALEtte
{NORMal|TEMPerature|SPECtral|INVErted}
ACQuire:FASTAcq:PALEtte?
```

Arguments `NORMal` colors traces according to their channel.

`TEMPerature` colors all traces using a multicolored palette, where “intensity” is represented by hue; blue for least frequently hit, red for most frequently hit. All traces share this palette. This is the default color palette.

`SPECtral` colors all traces using a multicolored palette, where “intensity” is represented by hue; red for least frequently hit, blue for most frequently hit. All traces share this palette.

`INVErted` Inverts the normal display hues and lightness levels based on sample intensity. The areas of lowest sample density appear the brightest, while the areas with the highest sample density appear the darkest.

Examples ACQUIRE:FASTACQ:PALETTE TEMPerature sets the acquisition fast acquisition palette to Temperature.

ACQUIRE:FASTACQ:PALETTE? might return :ACQUIRE:FASTACQ:PALETTE TEMPERATURE.

ACQuire:FASTAcq:STATE

Sets or queries the state of fast acquisition mode.

Group Acquisition

Syntax
```
ACQuire:FASTAcq:STATE {<NR1>|OFF|ON}
ACQuire:FASTAcq:STATE?
```

Arguments <NR1> = 0 disables FASTAcq; any other value turns this feature on.

OFF disables the FASTAcq feature.

ON enables the FASTAcq feature.

Examples ACQUIRE:FASTACQ:STATE ON turns on fast acquisition mode.

ACQUIRE:FASTACQ:STATE? might return :ACQUIRE:FASTACQ:STATE 0 indicating that fast acquisition mode is off.

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 reac

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

ON enables the stop acquisitions when a mask test fails feature.

OFF disables the stop acquisitions when a mask test fails feature.

<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.

Examples

ACTONEVent:MASKFail:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

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.

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

ON enables the save screen capture when a mask hit occurs feature.

OFF disables the save screen capture when a mask hit occurs feature.

<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.

Examples

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

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.

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 `ON` enables the generate an SRQ event when a mask hit occurs feature.

`OFF` disables the generate an SRQ event when a mask hit occurs feature.

`<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.

Examples `ACTONEVent:MASKHit:ACTION:SRQ:STATE 1` enables the generate an SRQ event when a mask hit occurs feature.

`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.

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 ON enables the stop acquisitions when a mask hit occurs feature.

OFF disables the stop acquisitions when a mask hit occurs feature.

<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.

Examples ACTONEVent:MASKHit:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

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.

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 ON enables the save source waveform(s) when a mask test passes feature.

OFF disables the save source waveform(s) when a mask test passes feature.

<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.

Examples ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

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.

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

ON enables the generate an SRQ event when a mask passes feature.

OFF disables the generate an SRQ event when a mask passes feature.

<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.

Examples

ACTONEVent:MASKPass:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask passes feature.

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.

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

ON enables the stop acquisitions when a mask test passes feature.

OFF disables the stop acquisitions when a mask test passes feature.

<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.

Examples

ACTONEVent:MASKPass:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

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.

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

```
ACTONEVent:SEARCH:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

`ON` enables the generate an SRQ event when any search event is found feature.

`OFF` disables the generate an SRQ event when any search event is found feature.

`<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.

Examples

`ACTONEVent:SEARCH:ACTION:SRQ:STATE 1` enables the generate an SRQ event when any search event is found feature.

ACTONEVent:SEARCH:ACTION:SRQ:STATE? might return :ACTONEVent:SEARCH:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when any search event is found feature is disabled.

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE

This command stops acquisitions when a search event is found.

Group Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

ON enables the stop acquisitions when a search event is found feature.

OFF disables the stop acquisitions when a search event is found feature.

<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.

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:HIGHLevel

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:HIGHLevel <NR3>
AFG:HIGHLevel?
```

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:LOWLevel <NR3>
AFG:LOWLevel?
```

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 NR3 is the 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 µs.

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 µs.

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 NR3 is a 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 NR3 is a 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
```
ALIas:DEFine <QString><,>{<QString>|<Block>}
ALIas:DEFine?
```

Related Commands ALIas:STATE

Arguments The first `<QString>` is the alias label.

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.

The second `<QString>` or `<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 256 characters.

Examples `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.

ALIAS:DEFINE? "ST1" returns :ALIAS:DEFINE "ST1",#247
:RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE.

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 ALIas:DELETe <QString>

Related Commands *ESR?

ALIas:DELETe:ALL

Arguments <QString> is the name of the alias to be removed. Using ALIas:DELETe without specifying an alias causes an execution error. <QString> must be a previously defined value.

Examples ALIAS:DELETE "SETUP1" 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
```
AUTOSet:TRIGger:ENAble {ON|1|OFF|0}
AUTOSet:TRIGger:ENAble?
```

Related Commands AUTOSet:ACQuisition:ENAble

AUTOSet:HORizontal:ENAble

AUTOSet:VERTical:ENAble

Arguments `ON` or `1` enables Autoset to change trigger settings.

`OFF` or `0` disables Autoset from changing trigger settings.

Examples

`AUTOSet:TRIGger:ENAble OFF` turns off Autoset's adjustment of the trigger settings.

`AUTOSet:TRIGger:ENAble?` might return 0, indicating Autoset's adjustment of the trigger settings is turned off.

AUTOSet:VERTical:ENAble

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

```
AUTOSet:VERTical:ENAble {ON|1|OFF|0}
AUTOSet:VERTical:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble

AUTOSet:HORizontal:ENAble

AUTOSet:TRIGger:ENAble

Arguments

`ON` or `1` enables Autoset to change vertical settings.

`OFF` or `0` disables Autoset from changing vertical settings.

Examples

`AUTOSet:VERTical:ENAble OFF` 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

```
BUS:B<x>:CAN:FD:BITRate:CUSTom <NR1>
BUS:B<x>:CAN:FD:BITRate:CUSTom?
```

Related Commands BUS:B<x>:CAN:BITRate

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

<NR1> is the custom FD bit rate for the specified bus.

Examples BUS:B1:CAN:FD:BITRATE:CUSTOM 1000000 sets the bit rate to 1,000,000.

BUS:B1:CAN:FD:BITRATE:CUSTOM? might return :BUS:B1:CAN:FD:BITRATE:CUSTOM 4000000, 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

```
BUS:B<x>:CAN:SAMPLEpoint <NR1>
BUS:B<x>:CAN:SAMPLEpoint?
```

Arguments

B<x> is the number of the bus.

<NR1> is the sample point, in percent, for the specified CAN bus.

Examples

`BUS:B1:CAN:SAMPLEpoint 40` sets the sample point to 40%.

`BUS:B1:CAN:SAMPLEpoint?` might return `:BUS:B1:CAN:SAMPLEPOINT 50` 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:SINal CANL` sets the signal type to CANL.

`BUS:B1:CAN:SINal?` 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
```
BUS:B<X>:CAN:THReshold <NR3>
BUS:B<X>:CAN:THReshold?
```

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

<NR3> is the source channel threshold for the specified CAN bus.

Examples BUS:B1:CAN:THReshold 5 sets the threshold to 5 V.

BUS:B1:CAN:THReshold? might return :BUS:B1:CAN:THRESHOLD 0.0E+0 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

```
BUS:B<x>:DISplay:FORMat
{HEX|BINARY|MIXEDASCII|MIXEDHEX|ASCII|DECIMAL|MIXED}
BUS:B<x>:DISplay:FORMat?
```

Arguments

B<x> 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

BUS:B1:DISplay:FORMat BINARY sets the format to binary.

BUS:B1:DISplay:FORMat? might return :BUS:B1:DISPLAY:FORMAT HEX 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

```
BUS:B<x>:I2C:DATa:THReshold <NR3>
BUS:B<x>:I2C:DATa:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the I2C Data (SDA) source threshold for the specified bus.

Examples

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

BUS:B1:I2C:DATa:THReshold? might return :BUS:B1:I2C:DATA:THRESHOLD 0.0E+0 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
```
BUS:B<x>:I2C:RWINADDR {0|1}
BUS:B<x>:I2C:RWINADDR?
```

Related Commands BUS:B<x>:I2C:CLOCk:SOUrce

BUS:B<x>:I2C:DATa:SOUrce

Arguments B<x> 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 BUS:B1:I2C:RWINADDR 0 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:LABel:FONT:BOLD OFF` turns off bold font.

`BUS:B1:LABel: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>:LABel:FONT:ITALic {ON|OFF|1|0}
BUS:B<x>:LABel: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:B1LABel:FONT:ITALic OFF turns off italic font.

BUS:B1LABel: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>:LABel:FONT:SIZE <NR1>
BUS:B<x>:LABel:FONT:SIZE?
```

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

<NR1> is the font size.

Examples BUS:B1:LABel:FONT:SIZE 10 sets the font size to 10 points.

BUS:B1:LABel:FONT:SIZE? might return :BUS:B1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

BUS:B<x>:LABel: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>:LABel:FONT:TYPE <QString>
BUS:B<x>:LABel: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:LABel:FONT:TYPE Monospace selects a monospace font.

BUS:B1:LABel:FONT:TYPE? might return :BUS:B1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman" indicating the font type is Frutiger LT Stdn.

BUS:B<x>:LABel: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>:LABel:FONT:UNDERline {ON|OFF|1|0}
BUS:B<x>:LABel: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:LABel:FONT:UNDERline ON turns on underline font.

BUS:B:LABel: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>:LABel:XPOS <NR3>
BUS:B<x>:LABel: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:LABel:XPOS 90 set the x position to 90.

BUS:B1:LABel:XPOS? might return :BUS:B1:LABEL:XPOS 45.0000 indicating the x position is 45.0 pixels to the right of the left edge of the display.

BUS:B<x>:LABel: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>:LABel:YPOS <NR3>
BUS:B<x>:LABel: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:LABel:YPOS 1.0e1 set the y position to 10.0.

BUS:B1:LABel:YPOS? might return :BUS:B1:LABEL: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:THReshold <NR3>
BUS:B<x>:LIN:SOUrce:THReshold?
```

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

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

Examples `BUS:B1:LIN:SOUrce:THReshold 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:LIN:SOUrce:THReshold?` 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>
BUS:B<x>:PARallel:ALLTHResholds?
```

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 x 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}
BUS:B<x>:PARallel:BIT<x>SOUrce?
```

Related Commands BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold

Arguments

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

`BIT<x>` is the number of the bit source.

`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<x>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 x and is an integer in the range of 1 to 64.

Group Bus

Syntax

```
BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold <NR3>
BUS:B<x>:PARallel:BIT<x>SOUrce:THReshold?
```

Related Commands

BUS:B<x>:PARallel:BIT<x>SOUrce

Arguments

B<x> is the number of the bus.

BIT<x> is the number of the bit source.

<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

B<x> is the number of the bus.

FALLING decodes on the falling edge of the clocked parallel bus signal.

RISING decodes on the rising edge of the clocked parallel bus signal.

EITHER decodes on the rising or falling edge of the clocked parallel bus signal.

Examples

BUS:B1:PARALLEL:CLOCK:EDGE FALLING sets the decoding to happen on the falling edge of its clocked parallel bus B1.

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.

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

```
BUS:B<x>:PARallel:CLOCk:ISCLOCKED {OFF|ON|<NR1>}
BUS:B<x>:PARallel:CLOCk:ISCLOCKED?
```

Arguments

B<x> is the number of the bus.

OFF argument specifies an asynchronous bus.

ON argument specifies a clocked bus.

<NR1> = 0 specifies an asynchronous bus; any other value specifies a clocked bus.

Examples

BUS:B1:PARallel:CLOCk:ISCLOCKED 0 sets the bus to operate asynchronously.

BUS:B1:PARallel:CLOCk:ISCLOCKED? might return :BUS:B1:PARALLEL:CLOCK:ISCLOCKED 1 indicating the bus is clocked.

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

```
BUS:B<x>:PARallel:CLOCkSOUrce {CH<x>| DCH<x>_D<x>| MATH<x>|
REF<x>| REF<x>_D<x>| NONE}
BUS:B<x>:PARallel:CLOCkSOUrce?
```

Related Commands BUS:B<x>:PARallel:CLOCk:ISCLOCKED

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog FlexChannel to use as the bus clock source.

DCH<x>_D<x> 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.

MATH<x> specifies the math channel to use as the bus clock source.

REF<x> specifies the reference channel to use as the bus clock source.

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

NONE specifies the reference channel to use as the bus clock source.

Examples

BUS:B1:PARALLEL:CLOCK:SOURCE CH3 sets the Parallel clock source for the bus B1 to channel 3.

BUS:B1:PARALLEL:CLOCK:SOURCE? might return :BUS:B1:PARALLEL:CLOCK:SOURCE CH8, indicating that the Parallel clock source for the bus B1 is set to channel 8.

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

```
BUS:B<x>:PARallel:CLOCkSOUrce:THReshold <NR3>
BUS:B<x>:PARallel:CLOCkSOUrce:THReshold?
```

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

```
BUS:B<x>:RS232C:BITRate:CUSTom <NR1>
BUS:B<x>:RS232C:BITRate:CUSTom?
```

Related Commands BUS:B<x>:RS232C:BITRate

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

<NR1> is the custom bit rate.

Examples BUS:B1:RS232C:BITRATE:CUSTOM 9500 sets the bit rate for the RS-232C bus B1 to 9500 bits-per-second.

BUS:B1:RS232C:BITRATE:CUSTOM? might return :BUS:B1:RS232C:BITRATE:CUSTOM 9500, 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
```
BUS:B<x>:RS232C:DATABits {<NR1>}
BUS:B<x>:RS232C:DATABits?
```

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

<NR1> specifies the number of bits in the RS-232C data frame.

Examples BUS:B1:RS232C:DATABITS 8 sets the data bits to 8 for the RS-232C bus B1.

BUS:B5:RS232C:DATABITS? might return :BUS:B1:RS232C:DATABITS 7, 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 {NORmal|INVERTed}
BUS:B<x>:RS232C:POLarity?
```

Arguments

B<x> is the number of the bus.

NORmal 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:THReshold <NR3>
BUS:B<x>:RS232C:SOUrce:THReshold?
```

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 to12 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 4nibbles

SIX specifies regular data with 6nibbles

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|NORmal}
BUS:B<x>:SENT:POLARITY?
```

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

INVerted specifies inverted polarity.

NORmal 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 B<x> is the number of the bus waveform.

NOne specifies no slow channel configured.

ENHANCED4 specifies Enhanced 4 slow channel configuration.

ENHANCED8 specifies Enhanced 8 slow channel configuration.

SHOrt specifies short slow channel configuration.

Examples BUS:B1:SENT:SLOW SHORT sets the SENT bus to use the short slow channel configuration.

BUS:B1:SENT:SLOW? might return :BUS:B1:SENT:SLOW NONE to indicate that the SENT bus is set to have no slow channel.

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

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

CH<x> specifies an analog channel as the clock source waveform for the audio bus.

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.

Math<x> specifies a math waveform as the clock source waveform for the audio bus.

REF<x> specifies a reference waveform as the clock source waveform for the audio bus

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
```
BUS:B<x>:SENT:TICKTIME <NR3>
BUS:B<x>:SENT:TICKTIME?
```

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

<NR3> is the SENT clock tick time, in seconds.

Examples BUS:B1:SENT:TICKTIME 3.0E-6 sets the SENT bus to clock tick to 3 microseconds.

BUS:B1:SENT:TICKTIME? might return :BUS:B1:SENT:TICKTIME 10.0000E-6 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
```
BUS:B<x>:SENT:TICKTOLerance <NR3>
BUS:B<x>:SENT:TICKTOLerance?
```

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

`<NR3>` is the tick tolerance percentage.

Examples `BUS:B1:SENT:TICKTOLERANCE 20.0` sets the tick tolerance to 20%.

`US:B1:SENT:TICKTOLERANCE?` might return
`:BUS:B1:SENT:TICKTOLERANCE 22.5` 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 `B<x>` is the number of the bus waveform.

`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.

`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.

Examples

`BUS:B1:SPI:BITORDER LSB` sets each bit order for the SPI Bus 1 to LSB.

`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.

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

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

`FALL` sets the clock to the falling edge of the signal.

`RISE` sets the clock to the rising edge of the signal.

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 µs.

`BUS:B1:SPI:IDLETime?` might return `BUS:B1:SPI:IDLETIME 5.0000E-6`, indicating that the idle time is set to 5 µs.

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

```
BUS:B<x>:SPI:MISo:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:MISo: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:MISo:DATa:POLarity HIGH sets the SPI MISo data polarity to high.

BUS:B1:SPI:MISo:DATa:POLarity? might return BUS:B1:SPI:MISo:DATa:POLarity HIGH, 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

```
BUS:B<x>:SPI:MISo:INPut {CH<x>| DCH<x>_D<x>| MATH<x>|
REF<x>}
BUS:B<x>:SPI:MISo:INPut?
```

Arguments

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

CH<x> 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 thresold 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:THReshold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:SPI:SELect:THReshold? 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:FACtory:STATus? (Query Only)

This query returns the factory calibration status value saved in nonvolatile memory.

Group Calibration

Syntax `CALibrate:FACtory:STATus?`

Examples `CALIBRATE:FACTORY:STATUS?` might return `CALIBRATE:FACTORY:STATUS PASS`, indicating the factory 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:DATE? (Query Only)

This query returns the date and time of the most recent signal path calibration.

Group Calibration

Syntax `CALibrate:INTERNal:DATE?`

Related Commands *CAL?

Returns This query returns the date and time (UTC) of the most recent signal path calibration.

Examples `CALIBRATE:INTERNAL:DATE?` might return `"3/10/2026 22:36:41"`, indicating the last signal path calibration ran was on March 10th, 2026.

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:ADDNew (No Query Form)

This command adds the specified callout. A Note is the default callout type.

Group Callouts

Syntax `CALLOUTS:ADDNew <QString>`

Arguments <QString> specifies the callout. The argument is of the form "CALLOUT<NR1>", where <NR1> is a number value ≥ 1.

Examples CALLOUTS:ADDNew "CALLOUT1" creates a new callout named CALLOUT1.

CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE

This command sets or queries the source of the Bookmark callout type.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE {CH1|CH2|CH3|CH4}
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE?
```

Arguments CH1, CH2, CH3, CH4 specifies the bookmark callout source. Select one.

Examples CALLOUTS:CALLOUT1:BOOKMark:SOURCE CH2 sets the bookmark callout source as Ch2 for callout1.

CALLOUTS:CALLOUT1:BOOKMark:SOURCE?, 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
```
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS <NR1>
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS?
```

Arguments <NR1> specifies the location of the bookmark linked to the source waveform in X-axis.

Examples CALLOUTS:CALLOUT1:BOOKMark:XPOS 9.28µs sets the bookmark callout X-Position to 9.28 µs for callout1.

CALLOUTS:CALLOUT1:BOOKMark:XPOS?, 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.

CALLOUTS:DELete (No Query Form)

This command deletes the specified callout. A Note is the default callout type.

Group Callouts

Syntax
```
CALLOUTS:DELete <QString>
```

Arguments

`<QString>` specifies the callout. The argument is of the form "CALLOUT<NR1>", where <NR1> is a number value ≥ 1.

Examples

`CALLOUTS:DELete "CALLOUT1"` deletes the callout named CALLOUT1.

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>|FULl}
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|DCREJect}
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>
CH<x>:DITHERrange?
```

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

CH1:DITHERrange 10 sets the amount of dithering as a percentage of full scale to 10.

CH1:DITHERrange? might return CH1:DITHERrange 10, which if vertical scale is 1V/div then the dithering amplitude will be 1 Vpk-pk.

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

`OFF` turns off the channel invert.

`ON` turns on the channel invert.

`<NR1>` 0 turns off the channel invert; any other value turns on the channel invert.

Examples

`CH2:INVert OFF` turns off the invert on channel 2.

`CH2:INVert?` might return `CH2:INVert 1`, indicating channel 2 is inverted.

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>
CH<x>:LABel:COLor?
```

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>}
CH<x>:LABel:FONT:BOLD?
```

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>}
CH<x>:LABel:FONT:ITALic?
```

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>
CH<x>:LABel:FONT:SIZE?
```

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>
CH<x>:LABel:FONT:TYPE?
```

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>}
CH<x>:LABel:FONT:UNDERline?
```

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>
CH<x>:POSition?
```

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>}
CH<x>:PROBEFunc:EXTUnits:STATE?
```

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>
CH<x>:SCAle?
```

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.

`<NR3>` 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 `CONNected:STATus?`

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:STATus?` might return `:CONNected:STATus 1`, 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 `CONNected:USAGe:TRack: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 tracking usage status of the Connected Scope Preference feature.

Examples CONNected:USAGe:TRack:REQUested:STATus 1
CONNected:SAVEPREFerences 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 CONNected:USAGe:TRack:STATus?

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.

CUSTOMTABle:ADDNew (No Query Form)

This command adds new custom results table. Only one result table is supported for CLR measurement.

Group Measurement

Syntax CUSTOMTABle:ADDNew <Qstring>

Arguments <Qstring> specifies the custom results table name.

Examples CUSTOMTABLE:ADDNEW "TABLE1" adds a new custom table named TABLE1.

CUSTOMTABle:DELete (No Query Form)

This command deletes the custom result(s) table that was added. Only one result table is supported for CLR measurement.

Group Measurement

Syntax `CUSTOMTABle:DELete <Qstring>`

Arguments `<Qstring>` specifies the custom results table name.

Examples `CUSTOMTABLE:DELETE "TABLE1"` deletes the new custom table named TABLE1.

CUSTOMTABle:LIST? (Query Only)

This command queries the list of custom result tables has been added. Only one result table is supported for CLR measurement.

Group Measurement

Syntax `CUSTOMTABle:LIST? <Qstring>`

Arguments `<Qstring>` specifies the custom results table name.

Examples `CUSTOMTABLE:LIST?` might return `:CUSTOMTABle:LIST TABLE1` , indicating there is one custom result table.

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

WFMOutpre?

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 isset 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 WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, and WFMOutpre:BYT_Or. Setting the DATa:ENGdg value causes the corresponding WFMOutpre 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

WFMOutpre:ENCdg

WFMOutpre:BN_Fmt

WFMOutpre: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. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = ASC ; :BN_Fmt = N/A ; :BYT_Or = N/A ; :BYT_NR = 1,2,4.

`RIBinary` specifies signed 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). The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RI ; :BYT_Or = MSB ; :BYT_NR = 1,2.

`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. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RP ; :BYT_Or = MSB ; :BYT_NR = 1,2.

`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. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = FP ; :BYT_Or = MSB ; :BYT_NR = 4.

`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. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RI ; :BYT_Or = LSB ; :BYT_NR = 1,2.

`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. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = RP ; :BYT_Or = LSB ; :BYT_NR = 1,2.

`SFPbinary` specifies floating point data in IBM PC format. The SFPbinary argument only works on math waveforms or ref waveforms saved from math waveforms. The following are the DATa and WFMOutpre parameter settings (separated by semicolons): :ENCdg = BIN ; :BN_Fmt = FP ; :BYT_Or = LSB ; :BYT_NR = 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 forth 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 bya 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:LABel:FONT:ITALic ON turns on italic font.

DCH1_D1:LABel:FONT:ITALic? might return :DCH1_D1:LABel:FONT:ITALic 0, indicating the font is not italic.

DCH<x>_D<x>:LABel: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>:LABel:FONT:SIZE <NR1>
DCH<x>_D<x>:LABel: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:LABel:FONT:SIZE 16 sets the font size to 16 points.

DCH1_D1:LABel:FONT:SIZE? might return :DCH1_D1:LABel:FONT:SIZE 20, indicating the font size is 20 points.

DCH<x>_D<x>:LABel: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>:LABel:FONT:TYPE <QString>
DCH<x>_D<x>:LABel: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

`Fail` - run until a failure is found, then halt.

`Once` - run through one loop.

`Always` - run forever.

`Onfail` - run until a failure is found, then loop on it.

`Ntimes` - run n number of loops.

Examples

`DIAG:LOOP:OPTION ALWAYS` sets loop option to `ALWAYS`.

`DIAG:LOOP:OPTION?` returns current looping option value.

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

`DIAG:LOOP:OPTION:NTIMES 2` sets diagnostics to loop 2 times.

`DIAG:LOOP:OPTION:NTIMES?` might return `DIAG:LOOP:OPTION:NTIMES 1` indicating diagnostics will run once.

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 one of the following desired diagnostic areas.

`ALL`

`IO`

`ANALOG`

`SYSTEM`

`ASIC`

`ACQ`

`SIGNAL`

`MEMORY`

Examples

`DIAG:SELECT IO` will select the `IO` diagnostic area.

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 `CH<x>` specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.

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

CH<x> specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.

Examples

DISplay:Ch1:NORMALColor COLOR13 sets the channel 2 normal waveform color to the value 13.

DISplay:REF3:NORMALColor? might return COLOR12, indicating that the normal waveform color setting for reference waveform 3 is 12.

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 ≠ 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}
DISplay:GLObal:B<x>:STATE?
```

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 ≠ 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

```
DISplay:GLObal:CH<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:CH<x>:STATE?
```

Arguments

<NR1> = 0 disables the display of the specified channel; any other value enables display of the channel.

ON enables display of the specified channel.

OFF disables display of the specified channel.

Examples

DISplay:GLObal:CH1:STATE OFF turns off the display of channel 1.

DISplay:GLObal:CH1:STATE? might return :DISPLAY:GLOBAL:CH1:STATE 0 indicating that channel 1 is not displayed.

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

```
DISplay:GLObal:DCH<x>:STATE {<NR1>|OFF|ON}
DISplay:GLObal:DCH<x>:STATE?
```

Arguments

DCH<x> = specifies the digital channel. The supported value is 1.

<NR1> = 0 disables the display of the specified digital channel; any other value enables display of the channel.

ON enables display of the specified digital channel.

OFF disables display of the specified digital channel.

Examples DISplay:GLObal:DCH1:STATE OFF turns off the display of digital channel 1.

DISplay:GLObal:DCH1:STATE? might return :DISPLAY:GLOBAL:DCH1:STATE 0 indicating that digital channel 1 is not displayed.

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 ≠ 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}
DISplay:GLObal:MATH<x>:STATE?
```

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 ≠ 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}
DISplay:GLObal:REF<x>:STATE?
```

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<x>: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<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:MATHFFTView<x>: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 (ΔY/Δ 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 ΔY/Δ 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:BPOSition

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:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition?
```

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:BPOSition 50e-3` sets the position to 0.005 vertical units.

`DISplay:MATHFFTView1:CURSor:HBArs:BPOSition?` 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:ONEOVERDELTATVALUE? (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:ONEOVERDELTATVALUE?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISplay:MATHFFTView1:CURSor:ONEOVERDELTATVALUE? might return DISPLAY:MATHFFTVIEW1:CURSOR:ONEOVERDELTATVALUE 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.

ON|1 enables the cursors.

OFF|0 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:CURSor:VBArs:AUNIts?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR: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:CURSor:VBArs:BPOSition 2.0e9` sets the cursor B position to 2.0 GHz.

`DISplay:MATHFFTView1:CURSor:VBArs:BPOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR: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:CURSor:VBArs:BUNIts?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR: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

```
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the waveform cursor B position in the specified plot view.

Examples

`DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition 2.0E+9` sets the cursor position to 2.0 GHz.

`DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:BPOSITION 250.0E+6`, indicating the waveform cursor B position is 250 MHz.

DISplay:MATHFFTView<x>:GRIDlines

This command sets or queries the grid lines setting for the specified Math-FFT view.

Group Display Control

Syntax

```
DISplay:MATHFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:MATHFFTView<x>:GRIDlines?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`HORizontal` specifies horizontal grid lines.

`VERTical` specifies vertical grid lines.

`BOTH` specifies both vertical and horizontal grid lines.

Examples `DISplay:MATHFFTView1:GRIDlines VERTICAL` specifies vertical grid lines.

`DISplay:MATHFFTView1:GRIDlines?` might return `:DISPLAY:MATHFFTVIEW1:GRIDLINES BOTH` indicating both horizontal and vertical grid lines are displayed.

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>}
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE?
```

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>
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO?
```

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 `OFF` disables the persistence aspect of the display.

`AUTO` automatically set the persistence.

`INFPersist` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`INFInite` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`VARPersist` sets a display mode where set pixels are gradually dimmed.

`CLEAR` resets the persist time count down and clears the display of acquired points.

Examples `DISPLAY:PERSISTENCE VARPERSIST` sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARpersist command.

`DISPLAY:PERSISTENCE?` might return `:DISPLAY:PERSISTENCE OFF`, indicating that the persistence aspect of the display is disabled.

DISplay:PERSistence: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:PERSistence:RESET`

Related Commands DISplay:PERSistence

DISplay:VARpersist

Examples `DISPLAY:PERSISTENCE:RESET` clears the display of persistence data.

DISplay:PLOTView<x>: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

```
DISplay:PLOTView<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:PLOTView<x>:CURSor:ROLOCATION?
```

Arguments

PLOTView<x> is the waveform plot number.

GRATICULE sets the plot cursor readouts to display as part of the cursors in the plot view.

BADGE removes the plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:PLOTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for plot 1 to display in the plot graticule.

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.

DISplay:PLOTView<x>:AUTOScale

This command sets or queries the enabled state of autoscale for the specified plot.

Group

Display Control

Syntax

```
DISplay:PLOTView<x>:AUTOScale {OFF|ON|<NR1>}
DISplay:PLOTView<x>:AUTOScale?
```

Arguments

PLOTView<x> is the Plot 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:PLOTView1:AUTOScale OFF turns off plot autoscale.

DISplay:PLOTView3:AUTOScale? might return :DISPLAY:PLOTVIEW3:AUTOSCALE 0, indicating the plot autoscale is off for plot 3.

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:ONEOVERDELTATVALUE? (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:ONEOVERDELTATVALUE?
```

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:ONEOVERDELTATVALUE?` might return `:DISPLAY:PLOTVIEW1:CURSOR:ONEOVERDELTATVALUE 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` indication 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}
DISplay:PLOTView<x>:XAXIS:SCALE?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`LINEAR` creates a plot with linear scales.

`LOG` creates a plot with logarithmic scales.

Examples

`DISplay:PLOTView1:XAXIS:SCALE LOG` creates a plot with a logarithmic horizontal scale.

`DISplay:PLOTView1:XAXIS:SCALE?` might return `:DISplay:PLOTView1:XAXIS:SCALE LINEAR` indicating plots will have linear scales.

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}
DISplay:PLOTView<x>:YAXIS:SCALE?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`LINEAR` specifies a linear vertical scale.

`LOG` specifies a logarithmic vertical scale.

Examples

`DISplay:PLOTView1:YAXIS:SCALE LINEAR` sets the plot to use a linear vertical scale.

DISplay:PLOTView1:YAXIS:SCALE? might return :DISPLAY:PLOTVIEW1:YAXIS:SCALE LOG indicating a logarithmic vertical scale is used.

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<x>: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<x>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:REFFFTView<x>:CURSor:ROLOCATION?
```

Arguments

PLOTView<x> is the Reference FFT plot number.

GRATICULE sets the Reference FFT plot cursor readouts to display as part of the cursors in the plot view.

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.

Examples

DISplay:REFFFTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for Reference FFT plot 1 to display in the plot graticule.

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.

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|<NR1>}
DISplay:REFFFTView<x>:AUTOScale?
```

Arguments

REFFFTView<x> is the plot number.

<NR1> = 0 disables auto-scale on the specified reffftview; any other value turns this feature on.

OFF|0 disables auto-scale on the specified reffftview.

ON|1 enables the specified channel on the specified Waveform View.

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 `HBArs` specifies horizontal bar cursors, which measure in vertical units.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`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.

`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.

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:BPOSition

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:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition?
```

Arguments <NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:HBArs:BPOSition -90` sets the B cursor position to -90.

`DISplay:REFFFTView5:CURSor:HBArs:BPOSition?` 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 `DISplay:REFFFTView<x>:CURSor:HBArs:DELTa?`

Returns Returns the delta V cursor readout value of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:HBArs:DELTa?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:DELTA 59.9867396737569` 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
```
DISplay:REFFFTView<x>:CURSor:MODe {INDEPENDENT|TRACK}
DISplay:REFFFTView<x>:CURSor:MODe?
```

Arguments `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.

`INDEPENDENT` allows independent adjustment of the two cursors.

Examples `DISplay:REFFFTView5:CURSor:MODe TRACK` sets the cursors to track together.

`DISplay:REFFFTView5:CURSor:MODe?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:MODE INDEPENDENT` indicating the cursors move independently.

DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE? (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:ONEOVERDELTATVALUE?`

Examples `DISplay:REFFFTView5:CURSor:ONEOVERDELTATVALUE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:ONEOVERDELTATVALUE 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|<NR1>}
DISplay:REFFFTView<x>:CURSor:STATE?
```

Arguments `<NR1>` = 0 specifies the cursor is not visible; any other value displays the cursor.

`OFF|0` specifies the cursor is not visible.

`ON|1` 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 curosr 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 gird lines.

DISplay:REFFFTView5:GRIDlines? might return :DISPLAY:REFFFTVIEW5:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are diplayed

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 {ON|OFF|0|1|<NR1>}
DISplay:REFFFTView<x>:REF:REF<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified reference; any other value turns this feature on.

OFF|0 disables the display the specified reference.

ON|1 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>}
DISplay:SELect:REFerence?
```

Arguments

NONE

REF<x> where x is the specified reference waveform.

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>}
DISplay:SELect:SOUrce?
```

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.

BUS<x> specifies a bus to use as the source.

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

PLOT<x> specifies a plot as the source.

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

Examples

`DISplay:SELect:SOUrce Bus1` selects Bus 1 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>}
DISplay:SELect:VIEW?
```

Arguments

`WAVEVIEW1`

`MATHFFT<x>`

`PLOTVIEW<x>`

`REFFFT<x>`

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>}
DISplay:SELect:WAVEView<x>:SOUrce?
```

Arguments

`CH<x>`

`MATH<x>`

`BUS<x>`

REF<x>

PLOT<x>

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 dows 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;AXPOSITION -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:MODE INDEPENDENT;ASOURCE AUTO;DDT 9.91E+37;ONEOVERDELTATVALUE 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

```
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?
```

Arguments

`AUTO` specifies auto as the cursor A source.

`CH<x>` specifies an analog channel to use as the cursor A source.

`BUS<x>` specifies a bus to use as the cursor A source.

`DCH<x>_DALL` specifies a digital channel to use as the cursor A source The supported digital channel value is 1.

`MATH<x>` specifies a math waveform to use as the cursor A source.

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

`PLOT<x>` specifies a plot as the cursor A source.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce CH1` sets the cursor 1 source to channel 1.

DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ASOURCE REF1 indicating the cursor A source is reference 1.

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

```
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?
```

Arguments

AUTO specifies auto as the cursor B source.

CH<x> specifies an analog channel to use as the cursor B source.

BUS<x> specifies a bus to use as the cursor B source.

DCH<x>_DALL specifies a digital channel to use as the cursor B source The supported digital channel value is 1.

MATH<x> specifies a math waveform to use as the cursor B source.

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

PLOT<x> specifies a plot as the cursor B source.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce CH1 sets the cursor B source to channel 1.

DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:BSOURCE REF1 indicating the cursor b source is reference 1.

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 `HBArs` specifies horizontal bar cursors, which measure in vertical units.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`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.

`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.

Examples `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION VBArs` selects the vertical bar cursor type for the specified Waveform View and cursor.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION HBArs indicating that the specified cursor is set to HBArs.

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

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition?
```

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 DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition 3.0e0 sets the cursor position to 3 V.

DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:APOSITION 2.9303448275862 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

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts?
```

Arguments <QString> is the cursor A vertical units of the specified cursor in the specified Waveform View.

Examples DISplay:WAVEView1:CURSor:CURSOR1:HBArs:AUNIts? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:AUNITS "V" 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 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.

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.

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

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition?
```

Arguments <NR3> is the horizontal cursor B 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 the specified screen cursor in the specified Waveform View 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: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

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition?
```

Arguments

<NR3> the vertical cursor B position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION 25.0E-3 sets the y position of the screen cursor of waveview1 to 25 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION -53.0000E-03 indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.

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 µs.

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 µs.

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 µs.

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 µs.

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 µs.

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 µs.

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:APOSITION -1.2667480236557E-6 indicating the position is 1.27 µs.

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

```
DISplay:WAVEView<x>:DCH<x>_DALL:VERTical:POSition <NR3>
DISplay:WAVEView<x>:DCH<x>_DALL: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:DCH1_DALL:VERTical:POSition 1.0e0 sets the position to 1 division.

DISplay:WAVEView1:DCH1_DALL:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:DCH1_DALL:VERTICAL:POSITION -3.4400 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

```
DISplay:WAVEView<x>:FILTer {SINX|LINear}
```

Arguments LINEAr specifies linear interpolation, where acquired points are connected with straight lines.

SINX specifies sin(x)/x interpolation, where acquired points are fit to a curve.

Examples DISplay:WAVEView1:FILTer SINX specifies sine-curve interpolation, when magnifying waveforms.

DISplay:WAVEView1:FILTer? might return :DISplay:WAVEView1:FILTer LINEAR, 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 <NR1> = 0 disables the specified reference in the specified Waveform View; any other value turns this feature on.

OFF disables the specified reference in the specified Waveform View.

ON enables the specified reference in the specified Waveform View.

Examples DISplay:WAVEView1:REF:REF1:STATE OFF disables the specified reference in the specified Waveform View.

DISplay:WAVEView1:REF:REF1:STATE? might return :DISPLAY:WAVEVIEW1:REF:REF1:STATE 1 indicating the specified reference in the specified Waveform View is on.

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 DISplay:WAVEView1:REF:REF1:VERTical:POSition -2.5e0 sets the position to -2.5 divisions.

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.

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 * vertical zoom factor. The bottom of the zoom window is +5 * 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 * vertical zoom factor. The bottom of the zoom window is +5 * 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 "labscope1" 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:IPADDress

ETHERnet:GATEWay:IPADDress

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

```
EVENT?
```

Related Commands

ALLEv?

*CLS

DESE

*ESE

*ESR?

EVMsg?

*SRE

*STB?

Examples

EVENT? might return :EVENT 110, 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 `EVMsg?`

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:CWDFILESystem: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:CWDFILESystem: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_0 00.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:CWDFILESystem: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:CWDcommand.

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 indicated 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 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| SETTO50| SINGleseq| TOUCHSCReen| TRIGMode| USER| VERTPOS| VERTSCALE}`

Arguments Arguments are the following instrument buttons: AUTOset| BUS| CH1<x>| CLEAR| DEFaultsetup| FORCetrig| GPKNOB1| GPKNOB2| HORZPOS| MATh| REF| RUNSTop| SETTO50| SINGleseq| TOUCHSCReen| TRIGMode| USER| VERTPOS| VERTSCALE

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 :ACQuire:MODe? query might return: SAMPLE

With :VERBose OFF and :HEADer OFF, the :ACQuire: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

```
HORizontal:DELay:MODe {OFF|ON|<NR1>}
HORizontal:DELay:MODe?
```

Related Commands HORizontal:POSition

Arguments OFF sets the Horizontal Delay Mode to off. This causes the HORizontal:POSition command to operate like the HORIZONTAL POSITION knob on the front panel.

ON sets the Horizontal Delay Mode to on. This causes the HORizontal:DELay:TIMe command to operate like the HORIZONTAL POSITION knob on the front panel.

<NR1> = 0 sets the Horizontal Delay Mode to off; any other value sets this mode to on.

Examples `HORIZONTAL:DELAY:MODE OFF` sets the Horizontal Delay Mode to off, allowing the horizontal position command to operate like the HORIZONTAL POSITION knob on the front panel.

`HORIZONTAL:DELAY:MODE?` might return `HORIZONTAL:DELAY:MODE OFF` indicating that the Horizontal Delay Mode is off and that the horizontal position command operates like the HORIZONTAL POSITION knob on the front panel.

HORizontal:DELay:TIMe

This command sets or queries the horizontal delay time that is used when delay mode is on.

Group Horizontal

Syntax
```
HORizontal:DELay:TIMe <NR3>
HORizontal:DELay:TIMe?
```

Related Commands HORizontal:POSition

Arguments `NR3` is the delay in seconds.

Examples `HORizontal:DELay:TIME 0.3` 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 `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.

`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.

Examples `HORIZONTAL:MODE AUTO` sets the horizontal mode to auto.

`HORIZONTAL:MODE?` might return `:HORIZONTAL:MODE MANUAL`, indicating that the horizontal mode is manual.

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
```
HORizontal:MODE:MANual:CONFIGure
{HORIZontalscale|RECORDLength}
HORizontal:MODE:MANual:CONFIGure?
```

Arguments `HORIZontalscale` will change when sample rate is adjusted.

`RECORDLength` will change when sample rate is adjusted.

Examples `HORIZONTAL:MODE:MANUAL:CONFIGURE RECORDLength` allows the record length be adjusted when sample rate is changed in Manual mode.

HORizontal:MODE:RECOrdlength

This command sets or queries the record length.

Group Horizontal

Syntax
```
HORizontal:MODE:RECOrdlength <NR1>
HORizontal:MODE:RECOrdlength?
```

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 µs 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 ≈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:Licen`
`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

*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.
```

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 {<NR1>|ON|OFF}`

Arguments `ON` sets the state of the right results readout bar to on.

`OFF` sets the state of the right results readout bar to off.

`<NR1>` 0 sets the state of the right results readout bar to off. Any number other than 0 sets the state of the right results readout bar to on.

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>:COUNT:SEG<x>: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>:COUNT:SEG<x>: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:CTHReshold

This command sets or queries the number of consecutive waveform violations needed for the specified mask test to change from PASS to FAIL.

Group Mask

Syntax

```
MASK:MASK<x>:TESt:CTHReshold <NR1>
MASK:MASK<x>:TESt:CTHReshold?
```

Arguments

MASK<x> specifies the mask test.

<NR1> specifies the threshold value.

Examples

MASK:MASK1:TESt:CTHRESHOLD 5 sets the number of consecutive waveform violations to 5 for mask test 1 to fail.

MASK:MASK1:TESt:CTHRESHOLD? might return :MASK:MASK1:TESt:CTHRESHOLD 5, indicating 5 consecutive waveform violations must occur on mask test 1 for mask test 1 to fail.

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:HABSolute

This command sets or queries the mask horizontal tolerance in absolute units of seconds. This value is used when the mask TTYPe is set to ABSolute and the UPDatenow command is sent.

Group Mask

Syntax

```
MASK:MASK<x>:TOLerance:HABSolute <NR3>
MASK:MASK<x>:TOLerance:HABSolute?
```

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow

MASK:MASK<x>:TTYPe

Arguments

MASK<x> specifies the mask number.

<NR3> is the tolerance in units of seconds. The maximum time is the equivalent of one horizontal graticule division.

Examples

MASK:MASK1:TOLerance:HABSolute 1.5e-9 sets the horizontal tolerance of Mask1 to 1.5 ns. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK2:TOLerance:HABSolute? might return :MASK:MASK2:TOLerance:HABSolute 3.75e-7, indicating that the horizontal tolerance of mask2 is 375 ns.

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:VABSolute

This command sets or queries the mask vertical tolerance in absolute units of the mask source (generally volts). This value is used when the mask TTYPe is set to ABSolute and the UPDatenow command is sent.

Group Mask

Syntax
```
MASK:MASK<x>:TOLerance:VABSolute <NR3>
MASK:MASK<x>:TOLerance:VABSolute?
```

Related Commands MASK:MASK<x>:TOLerance:UPDatenow

MASK:MASK<x>:TTYPe

Arguments `MASK<x>` specifies the mask number.

`<NR3>` is the tolerance in units of the mask source. Most traces in the waveform window have vertical units of volts, but other units such as amps or degrees are possible. The maximum value is the equivalent of one vertical graticule division.

Examples

`MASK:MASK1:TOLerance:VABSolute 1.5e-9` sets the vertical tolerance of Mask1 to 1.5 ns. The mask is not recalculated until an UPDatenow command is sent.

`MASK:MASK1:TOLerance:VABSolute?` might return `:MASK:MASK1:TOLerance:VABSolute 3.75e-7`, indicating that the vertical tolerance of mask2 is 375 ns.

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:MASK<x>:TTYPe

This command sets or queries the type of tolerance values used when defining a tolerance mask.

Group

Mask

Syntax

```
MASK:MASK<x>:TTYPe {SCReen|ABSolute}
MASK:MASK<x>:TTYPe?
```

Related Commands

MASK:MASK<x>:TOLerance:HABSolute

MASK:MASK<x>:TOLerance:HORizontal

MASK:MASK<x>:TOLerance:VABSolute

MASK:MASK<x>:TOLerance:VERTical

Arguments

MASK<x> specifies the mask number.

SCReen indicates that the horizontal and vertical mask tolerances are defined in relative units of graticule divisions. There are always 10 horizontal divisions and 10 vertical divisions on the scope waveform display. When the tolerance type is SCReen, the tolerance values set by the HORizontal and VERTical commands are used to define the tolerance mask when an UPDatenow command is sent.

ABSolute indicates that the horizontal and vertical mask tolerances are defined in the absolute units of the mask source

waveform. These units are normally seconds and volts respectively. When the tolerance type is ABSolute, the tolerance

values set by the HABSolute and VABSolute commands are used to define the tolerance mask when an UPDatenow

command is sent.

Examples

MASK:MASK1:TTYPE ABSOLUTE specifies that the horizontal and vertical tolerances for mask1 are specified in absolute s and v units.

MASK:MASK2:TTYPE ABSOLUTE? might return :MASK:MASK2:TTYPE SCReen, indicating that the horizontal and vertical tolerances for mask2 are defined in screen-relative graticule division units.

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

<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}
MATH:MATH<x>:AVG:MODE?
```

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>
MATH:MATH<x>:AVG:WEIGht?
```

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}
MATH:MATH<x>:FUNCtion?
```

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}
MATH:MATH<x>:INTERpolation?
```

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

NONE turns off math gating.

SCREEN turns on gating, using the left and right edges of the screen.

CURSor limits math to the portion of the waveform between the vertical bar cursors, even if they are off screen.

Examples

MATH:MATH3:GATING CURSOR sets the spectral math plot to be gated by the cursors.

MATH:MATH2:GATING? might return :MATH:MATH:2:GATING SCREEN which indicates the spectral math plot is gated by the screen.

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>
MATH:MATH<x>:LABel:COLor?
```

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 MATH:MATH3:LABEL:COLOR "GREEN" sets the Math 3 label color to green.

MATH:MATH1:LABEL:COLOR? might return :MATH:MATH1:LABEL:COLOR "BLUE" indicating the color of the Math1 label is blue.

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}
MATH:MATH<x>:LABel:FONT:BOLD?
```

Arguments <NR1> = 0 turns off bold, and any other integer turns on bold.

OFF turns off bold.

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}
MATH:MATH<x>:LABel:FONT:ITALic?
```

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>
MATH:MATH<x>:LABel:FONT:SIZE?
```

Arguments

<NR1> is the font size of the specified math label.

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>
MATH:MATH<x>:LABel:FONT:TYPE?
```

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}
MATH:MATH<x>:LABel:FONT:UNDERline?
```

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. 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.

When the MATH:MATH<x>:TYPE is set to BASIC, SOURCE1 and SOURCE2 can be used.

Group Math

Syntax

```
MATH:MATH<x>:SOUrce<x> {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:SOUrce<x>?
```

Related Commands MATH:MATH<x>:TYPe

MATH:MATH<x>:FUNCtion

Arguments `MATH<x>` specifies the math number.

`SOUrce<x>` specifies the source number.

`CH<x>` specifies the source as channel.

`MATH<x>` specifies the source as math.

`REF<x>` specifies the source as reference.

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:HORZ

This command sets or queries the horizontal display scale of the spectral math waveform. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:HORZ {LOG|LINEAr}
MATH:MATH<x>:SPECTral:HORZ?
```

Arguments LOG sets the SpectralMag units to log.

LINEAr sets the SpectralMag units to linear.

Examples MATH2:SPECTRAL:HORZ LOG sets the horizontal display scale of the spectral math waveform to log.

MATH2:SPECTRAL:HORZ? might return :MATH2:SPECTRAL:HORZ LINEAr, indicating the horizontal display scale of the Math 3 spectral math waveform is currently set to linear.

MATH:MATH<x>:SPECTral:MAG

This command sets or queries the units of the SpectralMag function in the specified math definition string. The Math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:MAG {LINEAr|DBM}
MATH:MATH<x>:SPECTral:MAG?
```

Arguments LINEAr sets the SpectralMag units to linear.

DBM sets the SpectralMag units to decibels. It also sets the Ref Level Offset to a value that is the equivalent of 1 mW into 50 Ω.

Examples MATH2:SPECTRAL:MAG DBM sets the SpectralMag units for Math 2 waveform to decibels.

MATH2:SPECTRAL:MAG? might return :MATH2:SPECTRAL:MAG DBM, indicating that the SpectralMag units for Math 2 waveform are set to decibels.

MATH:MATH<x>:SPECTral:PHASE

This command sets or queries the units of a SpectralPhase function in the specified math definition string. The Math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:PHASE {DEGrees| RADians| GROUPDelay}
MATH:MATH<x>:SPECTral:PHASE?
```

Arguments

DEGrees sets the SpectralPhase units to degrees.

RADians sets the SpectralPhase units to radians.

GROUPDelay sets the SpectralPhase units to groupdelay, which computes the derivative of unwrapped phase spectrum. Units are expressed in seconds.

Examples

MATH2:SPECTRAL:PHASE DEGREES sets the SpectralPhase units for Math 2 waveform to degrees.

MATH2:SPECTRAL:PHASE? might return :MATH2:SPECTRAL:PHASE RADIANS, indicating that the SpectralPhase units for Math 2 waveform are set to radians.

MATH:MATH<x>:SPECTral:SOUrce

This command sets or queries the specified spectral math source. This only works with a math of type FFT. The math waveform is specified by x. This command is for use when the MATH:MATH<x>:TYPE is FFT.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:SOUrce {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:SPECTral:SOUrce?
```

Related Commands

MATH:MATH<x>:TYPe

Arguments

CH<x> specifies the source as channel.

MATH<x> specifies the source as math.

REF<x> specifies the source as reference.

Examples

MATH:MATH3:SPECTRAL:SOURCE REF3 sets the source of the Spectral Math waveform to Ref 3.

MATH:MATH2:SPECTRAL:SOURCE? might return :MATH:MATH2:SPECTRAL:SOURCE CH4 indicating the Math 2 spectral source is Channel 4.

MATH:MATH<x>:SPECTral:SUPPress

This command sets or queries whether suppression threshold for the specified math waveform is enabled. This is only applied when Spectral Plot type is Phase. The math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:SUPPress {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:SUPPress?
```

Related Commands MATH:MATH<x>:SPECTral:TYPE

Arguments

0 disables suppression threshold for the specified math waveform.

1 enables suppression threshold for the specified math waveform.

OFF disables suppression threshold for the specified math waveform.

ON enables suppression threshold for the specified math waveform.

Examples

MATH:MATH3:SPECTRAL:SUPPRESS ON enables the suppression threshold on the Math 3 spectral waveform.

MATH:MATH2:SPECTRAL:SUPPRESS? might return :MATH:MATH2:SPECTRAL:SUPPRESS 0 indicating the suppression threshold is disabled on the Math 2 spectral waveform.

MATH:MATH<x>:SPECTral:SUPPress:VALue

This command sets or queries in volts the value of suppression threshold of the specified math waveform. This requires the Spectral type to be Phase and the Suppression to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:SUPPress:VALue <NR3>
MATH:MATH<x>:SPECTral:SUPPress:VALue?
```

Related Commands MATH:MATH<x>:SPECTral:TYPE

MATH:MATH<x>:SPECTral:SUPPress

Arguments <NR3> is the value of suppression threshold of the specified math waveform in volts.

Examples `MATH:MATH3:SPECTRAL:SUPPRESS:VALUE 100.0E-3` sets the suppression threshold of Math 3 to 100 mV.

`MATH:MATH2:SPECTRAL:SUPPRESS:VALUE?` might return `:MATH:MATH2:SPECTRAL:SUPPRESS:VALUE 10.0000E+30` indicating the suppression threshold of Math 2 is currently set to 10 kV.

MATH:MATH<x>:SPECTral:TYPE

This command sets or queries the FFT type selected for spectral analysis. The math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:TYPE {MAGNitude|PHASe|REAL|IMAGinary}
MATH:MATH<x>:SPECTral:TYPE?
```

Arguments

`MAGNitude` specifies the magnitude spectral function.

`PHASe` specifies the phase spectral function.

`REAL` specifies the real spectral function.

`IMAGinary` specifies the imaginary spectral function.

Examples `MATH2:SPECTRAL:TYPE REAL` specifies the real spectral function.

`MATH2:SPECTRAL:TYPE?` might return `:MATH2:SPECTRAL:TYPE MAGNITUDE`, indicating that the math is the magnitude spectral function.

MATH:MATH<x>:SPECTral:UNWRap

This command sets or queries whether phase unwrap of the spectral analyzer output data is enabled. The Math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:UNWRap {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:UNWRap?
```

Arguments

`0` disables phase unwrap for the specified math waveform.

`1` enables phase unwrap for the specified math waveform.

OFF disables phase unwrap for the specified math waveform.

ON enables phase unwrap for the specified math waveform.

Examples

MATH:MATH3:SPECTRAL:UNWRAP ON enables phase unwrap of the spectral analyzer output data.

MATH:MATH3:SPECTRAL:UNWRAP? might return :MATH:MATH3:SPECTRAL:SUPPRESS 0 indicating that the phase unwrap of the spectral analyzer output data is disabled.

MATH:MATH<x>:SPECTral:UNWRap:DEGrees

This command sets or queries how many degrees adjacent phase values can jump before being unwrapped. This requires the Spectral type to be Phase and the UNWRAP to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SPECTral:UNWRap:DEGrees <NR3>
MATH:MATH<x>:SPECTral:UNWRap:DEGrees?
```

Related Commands

MATH:MATH<x>:SPECTral:TYPE

MATH:MATH<x>:SPECTral:UNWRap

Arguments

<NR3> is the value of unwrap phase in degrees.

Examples

MATH:MATH3:SPECTRAL:UNWRAP:DEGREES 90 sets the unwrap phase of the spectral Math 3 to 90 degrees.

MATH:MATH3:SPECTRAL:UNWRAP:DEGREES? might return :MATH:MATH3:SPECTRAL:UNWRAP:DEGREES 180 indicating unwrap phase of the spectral math 3 waveform is 180 degrees.

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

`RECTANGular` window function is equivalent to multiplying all gate data by one.

`HAMMing` window function is based on a cosine series.

`HANNing` window function is based on a cosine series.

`BLACKMANHarris` window function is based on a cosine series.

`KAISERBessel` window function is based on a cosine series.

`GAUSSian` window function has the best localization characteristics in the joint time/frequency plane

`FLATTOP2` window function is a cosine series window with a flattened frequency response lobe.

`TEKEXPonential` window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

Examples

`MATH2:SPECTRAL:WINDOW GAUSSIAN` applies a Gaussian window to the spectral analyzer input data.

`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.

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

`DATa` sets the field type to DATa.

`MOSIdata` sets the field type to MOSIdata. MOSIdata field is available to select when SPI Bus configuration for Data Inputs is set to Two.

`MISOdata` sets the field type to MISOdata. MISOdata field is available to select when SPI Bus configuration for Data Inputs is set to Two.

Examples

`MATH:MATH2:SPI:SUPPortedfields DATa` sets the field type to DATa.

`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.

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}
MATH:MATH<x>:TYPe?
```

Arguments

`BASic` set the type to basic math.

`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

ADVanced. See examples for the command MATH:MATH<x>:DEFine.

`ADVanced` sets the type to advanced math.

Examples

`MATH:MATH2:TYPE BASIC` sets the type of Math 2 to basic.

`MATH:MATH2:TYPE?` might return `:MATH:MATH2:TYPE FFT` indicating the type of Math 4 is currently FFT.

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 `MATH:MATH2:VUNIT` "Small"

`MATH:MATH2:VUNIT?` might return `:MATH:MATH2:VUNIT` "Large" indicating the Math 2 vertical unit is set to "Large".

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.

MEASTABle:LIST? (Query Only)

This query-only command returns all measurement results view tables from the scope application.

Group Measurement

Syntax `MEASTABle:LIST?`

Returns Returns a list of all measurement results view tables.

Examples `MEASTABle:LIST "TABLE1, TABLE2"` indicating there are two measurement results view tables.

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 |NWIDTH |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 (R_M). Bursts are separated by a user-defined idle time (t_I). 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 (R_M) crossing on one source to a specified Mid reference level (R_M) 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 (R_T) to the Bottom reference level (R_B). 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 (R_T) to the Base reference level (R_B). 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 (R_T). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (R_M) on the Clock source to the closest specified Mid reference level (R_M) 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 (R_B). 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.

Negative Duty Cycle = (Negative Width) / Period × 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 (R_M). 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.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (R_M). 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.

Positive Duty Cycle = (Positive Width)/Period × 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 (R_M). 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.

Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (R_M). 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 (R_B) to the Top reference level (R_T). 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 (R_B) to the Top reference level (R_T). 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 (R_M) crossing on the Data source to the closest specified Mid reference level (R_M) 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 (R_M) crossing on one source to the following specified Mid reference level (R_M) 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 (R_T) and/or below the Base reference level (R_B). 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 ≥ 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

```
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid?
```

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:FALLMiD 50.0000` sets the MID reference level for the falling edge to 50%.

`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%.

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 measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the hysteresis value used for the autoset.

Examples `MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000` sets the reference level hysteresis to 5.0 mV.

`MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS?` might return `:MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the reference level hysteresis is set to 5.0 mV.

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}
MEASUrement:EDGE<x>?
```

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

CH<x>

MATH<x>

REF<x>

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 SEARCH<x>
MEASUrement:GATing:SEARCHSource?
```

Arguments

SEARCH<x> 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 <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 <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 <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:HYSTeresis

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:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments `<NR3>` is the hysteresis value used for the autoset.

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 `<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 `<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 `<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 `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

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 reverence 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

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.

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.

Examples

MEASUrement:MATH2:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly

MEASUrement:MATH2:REFLevels:METHod? might return :MEASUREMENT:MATH2:REFLEVELS:METHOD PERCENT indicating reference levels are in percent relative to HIGH and LOW.

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 <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.

Examples `MEASUrement:MATH2:REFLevels:PERCent:FALLHigh 95` sets the reference level to 95% of TOP.

`MEASUrement:MATH2:REFLevels:PERCent:FALLHigh?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 90.0000` indicating the reference level is set to 90% of TOP.

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

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow?
```

Arguments <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.

Examples `MEASUrement:MATH2:REFLevels:PERCent:FALLLow 5` sets the reference level to 5% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLLow? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference level is set to 10% of TOP.

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
```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid?
```

Arguments <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:FALLMid 50 sets the reference level to 50% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLMid? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the reference level is set to 50% of TOP.

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
```
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the hysteresis value used for the autoset.

Examples `MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis 2.5` sets the hysteresis to 2.5% of MAX

`MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is set to 5% of MAX.

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 <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 <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 <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

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:MATH2:REFLevels:PERCent:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:MATH2:REFLevels:PERCent:TYPE? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE TENNINETY indicating the reference levels are set to 10%, 50% and 90%.

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

RISE specifies a burst with a rising edge.

FALL specifies a burst with a falling edge.

Examples

MEASUrement:MEAS2:BURSTEDGTYPe FALL specifies a burst with a falling edge.

MEASUrement:MEAS2:BURSTEDGTYPe? might return :MEASUREMENT:MEAS2:BURSTEDGTYPE RISE indicating a burst with a rising edge.

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:MEAS2CCRESUlts: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>:CYCLemode

This command sets or queries the cycle mode for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:CYCLemode {RECORD|CYCLE|ROTATION}
MEASUrement:MEAS<x>:CYCLemode?
```

Related Commands

MEASUrement:MEAS<x>:TOEdge

Arguments

`MEAS<x>` specifies the measurement number.

`RECORD` specifies that the measurement is taken over the whole record.

`CYCLE` specifies that measurements are taken on each cycle of the source.

`ROTATION` specifies that measurements are taken on a rotation of the source.

Examples

`MEASUREMENT:MEAS1:CYCLemode CYCLE` sets the measurement to be taken over each cycle of the source.

`MEASUREMENT:MEAS1:CYCLemode?` might return `:MEASUREMENT:MEAS1:CYCLemode RECORD`, indicating the measurement is taken over the whole record.

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

```
MEASUrement:MEAS<x>:DELay:EDGE<x>
{FALL|RISe|BOTH|SAMEas|OPPositeas}
MEASUrement:MEAS<x>:DELay:EDGE<x>?
```

Related Commands

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: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.

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
```
MEASUrement:MEAS<x>:DISPlaystat:ENABle {OFF|ON|<NR1>}
MEASUrement:MEAS<x>:DISPlaystat:ENABle?
```

Arguments `MEAS<x>` specifies the measurement number.

`OFF` turns off the display of statistics in measurement badges.

`ON` turns on the display of statistics in measurement badges.

`<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}
MEASUrement:MEAS<x>:EDGE<x>?
```

Arguments

`MEAS<x>` specifies the measurement number.

`EDGE<x>` specifies the edge number.

`RISE` specifies the rising edge.

`FALL` specifies the falling edge.

`BOTH` specifies either the rising or falling edge.

Examples

`MEASUrement:MEAS1:EDGES1 RISE` specifies the edge 1 as the rising edge.

`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 measure 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:SAVEWAVEform: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:HYSTeresis

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:HYSTeresis <NR3>
MEASUrement:MEAS<x>:GATing:HYSTeresis?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the gating hysteresis.

Examples MEASUrement:MEAS1:GATing:HYSTeresis 25.0E-3 sets the hysteresis to 25 mV.

MEASUrement:MEAS1:GATing:HYSTeresis? might return :MEASUREMENT:MEAS1:GATING:HYSTERESIS 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>}
MEASUrement:MEAS<x>:GATing:LOGICSource?
```

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
MEASUrement:MEAS<x>:GATing:SEARCHSource?
```

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 µs.

`MEASUrement:MEAS1:IDLETime?` might return `:MEASUREMENT:MEAS1:IDLETIME 50.0000E-6` indicating the idle time is 50.0 µs.

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>
MEASUrement:MEAS<x>:LABel?
```

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

```
MEASUrement:MEAS<x>:PASSFAILHIGHlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILHIGHlimit?
```

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 high limit for a measurement test. The high limit is a number which a measurement result will be tested against.

Examples MEASUrement:MEAS1:PASSFAILHIGHlimit 710 specifies 710 as the high limit for measurement 1.

MEASUrement:MEAS1:PASSFAILHIGHlimit? might return :MEASUrement:MEAS1:PASSFAILHIGHlimit 710 indicating that 710 is the high limit for measurement 1.

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

```
MEASUrement:MEAS<x>:PASSFAILLIMit <NR2>
MEASUrement:MEAS<x>:PASSFAILLIMit?
```

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 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>:PASSFAILLOWlimit

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>:PASSFAILLOWlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILLOWlimit?
```

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 low limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples `MEASUrement:MEAS1:PASSFAILLOWlimit 690` specifies 690 as the limit for measurement 1.

`MEASUrement:MEAS1:PASSFAILLOWlimit?` might return `:MEASUrement:MEAS1:PASSFAILLOWlimit 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 that the given limit.

MEASUrement:MEAS1:PASSFAILWHEN? might return :MEASUrement:MEAS1:PASSFAILWHEN LESSthan indicating that the condition for measurement 1 test failure as less that 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}
MEASUrement:MEAS<x>:PERFREQ:EDGE?
```

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 reverence 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 MEAS<x> specifies the measurement number.

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.

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.

Examples MEASUREMENT:MEAS1:REFLEVELS1:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.

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.

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 MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level.

Examples MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh 95.0 sets the high reference level of the falling edge to 95%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 90.0000 indicating the high reference level is set to 90%.

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 MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level.

Examples MEASUrement:MEAS1:REFLevels1:PERCent:FALLLow 5.0 sets the low reference level of the falling edge is 5%.

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%.

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

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level.

Examples `MEASUrement:MEAS1:REFLevels1:PERCent:FALLMid 50.0` sets the mid reference level of the falling edge to 50%.

`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%.

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
```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the hysteresis value used for the autoset.

Examples `MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis 5.0` sets the hysteresis level to 5%.

`MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5.0%.

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 MEAS<x> specifies the measurement number.

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:MEAS1:REFLevels1:PERCent:TYPE TWENtyeighty sets the reference levels percent to 20%, 50% and 80%.

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

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

`MEAS<x>` specifies the measurement number.

`AUTO` sets the reference level for the measurement automatically.

`MANual` allows the user to set the reference level for the measurement.

Examples

`MEASUrement:MEAS1:REFMode Manual` allows the user to set the reference level for the measurement.

`MEASUrement:MEAS1:REFMode?` might return `:MEASUREMENT:MEAS1:REFMODE AUTO` indicating the reference levels for the measurement are set automatically.

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 `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum?`

Returns The maximum value found for the specified measurement since the last statistical reset.

Examples `MEASUrement:MEAS1:RESUlts:CURRentacq:MAXimum?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM 2.280` 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 `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN?`

Arguments MEAS<x> 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 `MEASUrement:MEAS3:RESUlts:CURRentacq:MEAN?` might return `:MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN 2.30` 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 `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum?`

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

MEAS<x> specifies the measurement number.

SOUrce<x> specifies the source number.

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.

Examples

MEASUrement:MEAS1:SOUrce1 CH1 sets the source for measurement 1 to channel 1.

MEASUrement:MEAS1:SOUrce1? might return :MEASUrement:MEAS1:SOUrce CH1 indicating the measurement source is channel 1.

MEASUrement:MEAS<x>:STATUS? (Query Only)

This command returns the pass fail status, if applicable, for the selected measurement. Measur are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:STATUS?
```

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

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}
MEASUrement:MEAS<x>:TRANSition?
```

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 specifyingthe 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.

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 (R_M). Bursts are separated by a user-defined idle time (t_I). 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 (R_M) crossing on one source to a specified Mid reference level (R_M) 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 (R_T) to the Bottom reference level (R_B). 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 (R_T) to the Base reference level (R_B). 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 (R_T). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (R_M) on the Clock source to the closest specified Mid reference level (R_M) 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 (R_B). 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.

Negative Duty Cycle = (Negative Width) / Period × 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 (R_M). 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.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (R_M). 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.

Positive Duty Cycle = (Positive Width)/Period × 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 (R_M). 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.

Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (R_M). 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 (R_B) to the Top reference level (R_T). 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 (R_B) to the Top reference level (R_T). 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 (R_M) 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 (R_M) crossing on one source to the following specified Mid reference level (R_M) 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 (R_T) and/or below the Base reference level (R_B). 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 MEASUrement:REFLevels:ABSolute:FALLHigh 1.5 sets the high reference level to 1.5 V.

MEASUrement:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 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

```
MEASUrement:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:REFLevels:ABSolute:FALLLow?
```

Arguments <NR3> is the value used as the low reference level of the falling edge.

Examples `MEASUrement:REFLevels:ABSolute:FALLLow -1.5` sets the low reference level to -1.5 V.

`MEASUrement:REFLevels:ABSolute:FALLLow?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.0000` 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

```
MEASUrement:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:REFLevels:ABSolute:FALLMid?
```

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: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.

Group Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REFLevels:ABSolute:HYSTeresis?
```

Arguments <NR3> is the value of the hysteresis of the reference level.

Examples `MEASUrement:REFLevels:ABSolute:HYSTeresis 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUrement:REFLevels:ABSolute:HYSTeresis?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:HYSTERESIS 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, as follows: `AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram`.

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

`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.

`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.

Examples

`MEASUrement:REFLevels:METHod ABSOLUTE` specifies that the reference levels are set explicitly.

`MEASUrement:REFLevels:METHod?` might return `:MEASUREMENT:REFLEVELS:METHOD PERCENT` indicating the reference levels are calculated as a percent relative to HIGH and LOW.

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

`LATCh` calculates reference levels only on the first acquisition after a statistics reset.

`CONTinuous` calculates reference levels on every acquisition.

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: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.

Group Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the percentage used to calculate the hysteresis of the reference level.

Examples `MEASUrement:REFLevels:PERCent:HYSTeresis 3` sets the hysteresis to 3%.

`MEASUrement:REFLevels:PERCent:HYSTeresis?` might return `:MEASUREMENT:REFLEVELS:PERCENT:HYSTERESIS 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: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.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis?
```

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:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.

MEASUrement:REF1:REFLevels:ABSolute:HYSTeresis? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYSTERESIS 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 {AUTO| MINMax|
MEANhistogram| MODEhistogram| EYEhistogram}
MEASUrement:REF<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.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:REF1:REFLevels:BASETop MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:REF1:REFLevels:BASETop? might return :MEASUREMENT:REF1:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:REF<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REF<x>:REFLevels:METHod?
```

Arguments

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.

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.

Examples

MEASUrement:REF1:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly.

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.

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: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.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the percentage used to calculate the hysteresis of the reference level.

Examples `MEASUrement:REF1:REFLevels:PERCent:HYSTeresis 2` sets the hysteresis to 2%.

`MEASUrement:REF1:REFLevels:PERCent:HYSTeresis?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:HYSTERESIS 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

```
MEASUrement:STATIstics:CYCLEMode {OFF|ON|0|1}
MEASUrement:STATIstics:CYCLEMode?
```

Arguments

`OFF` turns off statistics for all measurements. This is the default value.

`ON` turns on statistics and displays all statistics for each measurement.

`0` turns off statistics for all measurements.

`1` turns on statistics and displays all statistics for each measurement.

Examples

`MEASUREMENT:STATISTICS:CYCLEMODE OFF` turns off statistics for all measurements.

`MEASUREMENT:STATISTICS:CYCLEMODE`? might return `:MEASUREMENT:STATISTICS:CYCLEMODE 1` indicating that statistics are displayed for each measurement.

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

```
NEWpass <QString>
```

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 Continuos 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 Continuos 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 Continuos 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

```
PG:BIT:ZERO {HIGH|LOW|TOGGLE|HIGH_Z}
PG:BIT:ZERO?
```

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 Continuos mode.

Examples

PG:BIT:ZERO TOGGLE sets the Pattern Generator bit 0 output to toggle state.

PG:BIT:ZERO? might return :PG:BIT:ZERO HIGH, indicating the Pattern Generator bit 0 output is set to High.

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

```
PG:BITRate <NR3>
PG:BITRate?
```

Arguments

<NR3> 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

PG:BITRate 1000 sets the Pattern Generator bit rate to 1000 b/s.

PG:BITRate? might return :PG:BITRate 1, indicating the Pattern Generator bit rate is 1 b/s.

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

```
PG:BURSt:CCOUnt <NR1>
PG:BURSt:CCOUnt?
```

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 PG:BURSt:CCOUnt 1 sets the cycle count for Pattern Generator burst mode to 1.

PG:BURSt:CCOUnt? might return :PG:BURSt:CCOUnt 1, 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
```
PG:BURSt:TRIGger
```

Related Commands PG:BURSt:CCOUnt

Examples PG:BURSt:TRIGger 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
```
PG:FILE:PATTern <QString>
PG:FILE:PATTern?
```

Arguments <QString> is the file path that specifies the location of the specified instrument 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 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

`PG:FILE:PATTern "E:/2Power.csv"` specifies the path to the pattern file.

`PG:FILE:PATTern?` might return `PG:FILE:PATTern "E:/2Power.csv"`, indicating pattern file path.

PG:OUTPut:MODe

This command sets or queries the Pattern Generator output mode.

Conditions

Requires option 2-SOURCE.

Group

Pattern Generator

Syntax

```
PG:OUTPut:MODe {CONTinuous|BURSt|OFF}
PG:OUTPut:MODe?
```

Arguments

`CONTinuous` sets the Pattern Generator output type to continuous.

`BURSt` sets the Pattern Generator output type to burst.

`OFF` sets the Pattern Generator output type to off. This is the default value.

Examples

`PG:OUTPut:MODe BURSt` sets the Pattern Generator mode to burst.

`PG:OUTPut:MODe?` might return `PG:OUTPut:MODe CONTinuous`, indicating the Pattern Generator mode is set to continuous.

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 {MANual|FILE}
PG:PATTERNdefinition?
```

Arguments `MANual` 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 MANual`, 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>` ≥ 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.

POWer:ADDNew (No Query Form)

This command adds the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax `POWer:ADDNew "POWER<x>"`

Arguments "POWER<x>" is the Power Measurement Badge and must be POWER1.

Examples `POWER:ADDNEW "POWER1"` adds POWER1 measurement badge and selects Switching CLR by default.

POWer:DELete (No Query Form)

This command deletes the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax `POWer:DELete "POWER<x>"`

Arguments "`POWER<x>`" is the Power Measurement Badge and must be POWER1.

Examples `POWER:DELETE "POWER1"` deletes the POWER1 measurement badge.

POWer:POWer<x>:CLRESPONSE:AMP<x>Val

This command sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax
```
POWer:POWer<x>:CLRESPONSE:AMP<x>Val <NR3>
POWer:POWer<x>:CLRESPONSE:AMP<x>Val?
```

Arguments `POWer<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

`AMP<x>` sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.

`<NR3>` sets the generator amplitude for the specified configuration step, in the range of –10 mV to 100 V.

Examples `POWer:POWer1:CLRESPONSE:AMP3Val 20` sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

`POWer:POWer1:CLRESPONSE:AMP8Val?` might return `60`, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWer:POWer<x>:CLRESPONSE:AMPMode

This command sets or queries the amplitude mode for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:CLRESPONSE:AMPMode?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

`CONSTant` sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

`PROFile` enables configuring the generator to set amplitude values for each frequency band.

Examples

`POWer:POWer1:CLRESPONSE:AMPMode CONSTant` sets the amplitude mode to Constant for power measurement 1.

`POWer:POWer1:CLRESPONSE:AMPMode?` might return `PROFile`, indicating that the amplitude mode power measurement 1 is set to Profile.

POWer:POWer<x>:CLRESPONSE:ANALYSISMethod? (Query Only)

This command sets or queries the Analysis Method for Control Loop Response measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:ANALYSISMethod?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

Examples

`POWer:POWer1:CLRESPONSE:ANALYSISMethod?` might return `:POWer:POWer1:CLRESPONSE:ANALYSISMethod FFT`, indicating that the Analysis Method is FFT.

POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? (Query Only)

This command queries connection status to the external generator used with the specified Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?
```

Related Commands POWer:POWer<x>:CLRESPONSE:GENIPADDress

POWer:POWer<x>:CLRESPONSE:GENerator

Arguments `POWer<x>` is the number of the power measurement. This must be POWer1.

Returns `SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples `POWer:POWer1:CLRESPONSE:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude

This command sets or queries the constant amplitude voltage for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude <NR3>
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

<NR3> is the constant amplitude voltage value for the measurement, in the range of 10 mV to 100 V.

Examples

POWer:POWer1:CLRESPONSE:CONSTAMPlitude 50 sets the constant amplitude voltage for Power measurement 1 to be 50 V.

POWer:POWer1:CLRESPONSE:CONSTAMPlitude? might return 5.0E-3, indicating that the constant amplitude voltage for power measurement 1 is 50 mV.

POWer:POWer<x>:CLRESPONSE:FREQ<x>Val

This command sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.

Conditions

Requires option 2-SOURCE or the 2-ULTIMATE

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val <NR3>
POWer:POWer<x>:CLRESPONSE:FREQ<x>Val?
```

Arguments

POWer<x> sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

FREQ<x> sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

<NR3> sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

POWer:POWer1:CLRESPONSE:FREQ1Val 200 sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 1.

POWer:POWer1:CLRESPONSE:FREQ3Val? might return 2.000E+6, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 1.

POWer:POWer<x>:CLRESPONSE:GENerator

This command sets or queries the generator source for the specified Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENerator {INTernal|EXTernal}
POWer:POWer<x>:CLRESPONSE:GENerator?
```

Arguments

`POWer<x>` is the number of the power measurement. This must be POWer1.

`INTernal` sets the internal generator as the source for the Control Loop Response power measurement.

`EXTernal` sets the external generator as the source for the Control Loop Response power measurement.

Examples

`POWer:POWer1:CLRESPONSE:GENerator Internal` sets the internal generator as the source for the Control Loop Response measurement.

`POWer:POWer1:CLRESPONSE:GENerator?` might return `EXTERNAL`, indicating that the Control Loop Response measurement is set to use an external generator.

POWer:POWer<x>:CLRESPONSE:GENIPADDress

This command sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENIPADDress <NR2>
POWer:POWer<x>:CLRESPONSE:GENIPADDress?
```

Arguments

`POWer<x>` is the number of the power measurement. This must be POWer1.

`<NR2>` is the IP address of the generator.

Examples

`POWer:POWer1:CLRESPONSE:GENIPADDress 127.1.1.1` sets the IP address of the external generator to 127.1.1.1.

`POWer:POWer1:CLRESPONSE:GENIPADDress?` might return "", indicating that there is no address set for the external generator associated with Control Loop Response power measurement 4.

POWer:POWer<x>:CLRESPONSE:IMPEDance

This command sets or queries the vertical termination impedance for the Control Loop Response power measurement.

Conditions

Requires option 2-SOURCE or the 2-ULTIMATE

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:IMPEDance {FIFTy|HIGHZ}
POWer:POWer<x>:CLRESPONSE:IMPEDance?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

`FIFTy` sets the impedance to be 50 Ω.

`HIGHZ` sets the impedance to be 1 MΩ.

Examples

`POWer:POWer1:CLRESPONSE:IMPEDance FIFTy` sets the vertical termination impedance for power measurement 1 to be 50 Ω.

`POWer:POWer1:CLRESPONSE:IMPEDance?` might return `HIGHZ`, indicating that the vertical termination impedance for power measurement 1 is 1 MΩ.

POWer:POWer<x>:CLRESPONSE:INPUTSOurce

This command sets or queries the input source for the Control Loop Response power measurement.

Conditions

Requires option 2-SOURCE or the 2-ULTIMATE

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:INPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

CH<x> sets the channel to use for the output signal source.

Examples

POWer:POWer1:CLRESPONSE:INPUTSOurce CH1 sets the input for power measurement 1 to be the channel 1 waveform.

POWer:POWer1:CLRESPONSE:INPUTSOurce? might return CH1, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce

This command sets or queries the output source for the Control Loop Response power measurement.

Conditions

Requires option 2-SOURCE or the 2-ULTIMATE

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

CH<x> sets the channel to use for the output signal source.

Examples

POWer:POWer1:CLRESPONSE:OUTPUTSOurce CH1 sets the output source for power measurement 1 to be the channel 1 waveform.

POWer:POWer1:CLRESPONSE:OUTPUTSOurce? might return CH1, indicating that the output source for power measurement 1 is channel 1.

POWer:POWer<x>:CLRESPONSE:PPD

This command sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:PPD <NR3>
POWer:POWer<x>:CLRESPONSE:PPD?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

<NR3> is the PPD value for the measurement, in the range of 10 to 100 points.

Examples POWer:POWer1:CLRESPONSE:PPD 30 sets the PPD for Power measurement 1 to be 30 points.

POWer:POWer1:CLRESPONSE:PPD? might return 20, indicating that the PPD value for power measurement 1 is 20 points.

POWer:POWer<x>:CLRESPONSE:STARTFREQuency

This command sets or queries the start frequency value for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:STARTFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STARTFREQuency?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

<NR3> is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples POWer:POWer1:CLRESPONSE:STARTFREQuency 12 sets the starting frequency for power measurement 1 to be 12 Hz.

POWer:POWer1:CLRESPONSE:STARTFREQuency? might return 2400, indicating that the starting frequency for power measurement 1 is 2400 Hz.

POWer:POWer<x>:CLRESPONSE:STOPFREQuency

This command sets or queries the stop frequency value for the Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:STOPFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STOPFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

`<NR3>` is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer1:CLRESPONSE:STOPFREQuency 120` sets the stop frequency for the control loop response power measurement 1 to be 120 Hz.

`POWer:POWer1:CLRESPONSE:STOPFREQuency?` might return `2000`, indicating that the stop frequency for the control loop response power measurement 1 is 2000 Hz.

POWer:POWer<x>:CLRESPONSE:TESTCONNection (No Query Form)

This command tests the connection to the external generator used with the specified Control Loop Response power measurement.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:TESTCONNection EXECute
```

Related Commands POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?

POWer:POWer<x>:CLRESPONSE:GENIPADDress

Arguments POWer<x> is the number of the PSRR power measurement. This must be POWer1.

EXECute runs the test connection function.

Examples POWer:POWer1:CLRESPONSE:TESTCONNection EXECute runs the test connection function for Control Loop Response power measurement 1.

POWer:POWer<x>:PRESET (No Query Form)

This command runs a power preset action for the specified power measurement number.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax POWer:POWer<x>:PRESET {EXECute}

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

EXECute runs the power preset action.

Examples POWer:POWer1:PRESET Execute runs the power preset action for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:MAXimum? (Query Only)

This command queries the maximum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax POWer:POWer<x>:RESUlts:CURRentacq:MAXimum? {PM| GAINCROSSOVERFREQ| GM| PHASECROSSOVERFREQ}

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

<QString> is the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

"PM" returns the Phase margin value.

"GAINCROSSOVERFREQ" returns the Gain crossover frequency value.

"GM" returns the Gain margin value.

"PHASECROSSOVERFREQ" returns the Phase cross over frequency value.

Examples

POWer:POWer1:RESUlts:CURRentacq:MAXimum? "PM" might return 32.8, indicating the maximum value of Phase margin for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:MEAN? (Query Only)

This command queries the mean value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions

Requires option 2-SOURCE or the 2-ULTIMATE

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MEAN? {PM|
GAINCROSSOVERFREQ| GM| PHASECROSSOVERFREQ}
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

<QString> is the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

"PM" returns the Phase margin value.

"GAINCROSSOVERFREQ" returns the Gain crossover frequency value.

"GM" returns the Gain margin value.

"PHASECROSSOVERFREQ" returns the Phase cross over frequency value.

Examples `POWer:POWer1:RESUlts:CURRentacq:MEAN? "PM"` might return `32.8`, indicating the mean value of Phase margin for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This command queries the minimum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions Requires option 2-SOURCE or the 2-ULTIMATE

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MINimum? {PM|
GAINCROSSOVERFREQ| GM| PHASECROSSOVERFREQ}
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge. This must be POWer1.

`<QString>` is the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

`"PM"` returns the Phase margin value.

`"GAINCROSSOVERFREQ"` returns the Gain crossover frequency value.

`"GM"` returns the Gain margin value.

`"PHASECROSSOVERFREQ"` returns the Phase cross over frequency value.

Examples `POWer:POWer1:RESUlts:CURRentacq:MINImum? "PM"` might return `32.8`, indicating the minimum value of Phase margin for the current acquisition.

*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 <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.

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.

ON sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

Examples *PSC 0 sets the power-on status clear flag to false.

*PSC? might return 1 to indicate that the power-on status clear flag is set to true.

*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 MASK<x> is the destination mask.

<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 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 `<file_path>` is the file path that specifies the location of 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 FILESystem:CWD) unless a complete path is specified:

- 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 ≥ 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>
REF:REF<x>:DESKew?
```

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>
REF:REF<x>:LABel:COLor?
```

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}
REF:REF<x>:LABel:FONT:BOLD?
```

Arguments

<NR1> = 0 disables bold font; any other value turns this feature on.

OFF disables bold font.

ON enables bold font.

Examples

REF:REF1:LABel:FONT:BOLD ON turns on the bold font.

REF:REF1:LABel:FONT:BOLD? might return :REF:REF1:LABEL:FONT:BOLD 0 indicating the bold font is off.

REF:REF<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified reference label.

Group Vertical

Syntax

```
REF:REF<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}
REF:REF<x>:LABel:FONT:ITALic?
```

Arguments

<NR1> = 0 disables italic font; any other value turns this feature on.

OFF disables italic font.

ON enables italic font.

Examples

REF:REF1:LABel:FONT:ITALic ON turns on the italic font.

REF:REF1:LABel:FONT:ITALic? might return :REF:REF1:LABEL:FONT:ITALIC 0 indicating the italic font is off.

REF:REF<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified reference label.

Group Vertical

Syntax

```
REF:REF<x>:LABel:FONT:SIZE <NR1>
REF:REF<x>:LABel:FONT:SIZE?
```

Arguments <NR1> is the font size of the label.

Examples REF:REF1:LABel:FONT:SIZE 20 sets the font size to 20 points.

REF:REF1:LABel: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>
REF:REF<x>:LABel:FONT:TYPE?
```

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}
REF:REF<x>:LABel:FONT:UNDERline?
```

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>:LABel:YPOS <NR1>
REF:REF<x>:LABel: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>
REF:REF<x>:SOUrce?
```

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 `:WFMInpre` 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

RECAll: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:CWDFILESystem: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:/Dut12tests.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 {NORMal|INVErted}
SAVe:IMAGe:COMPosition?
```

Arguments `NORMal` Sets the saved screen capture to Normal colors.

`INVErted` sets the saved screen capture to Inverted colors.

Examples `SAVE:IMAGE:COMPOSITION NORMal` 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:MASK (No Query Form)

Saves the given Waveview Mask to the specified file. Use the format [<path>]"<filename><ext>" for the argument. Specifying a path is optional. If no path is entered, the file is saved to the current working directory set by FILESystem:CWD. <path> uses the form "<drive>/<dir>.../". The relative path specification is: 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. The complete path specification is: 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.

Group Save and Recall

Syntax
```
SAVe:MASK <Qstring>
```

Related Commands FILESystem:CWD

RECAll:MASK

Arguments `Qstring` is a quoted string that defines the path and file name used to save the specified file, in the format [<path>] "<filename><.ext>"

Examples `SAVE:MASK MASK2,"DataMask.xml"` saves the mask2 segment mask to the designated file in the current working directory.

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 `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.

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.

To export an eye diagram plot data to a .csv file, the prerequisite command is MEASUrement:ADDMEAS TIE.

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

```
SAVe:REPOrt:COMMents <QString>
SAVe:REPOrt:COMMents?
```

Arguments <QString> is the comments to be included in saved report files.

Examples SAVE:REPORT:COMMENTS "Test 3" adds comments to the report.

SAVE:REPORT:COMMENTS? might return :SAVE:REPORT:COMMENTS "High Temp Test 1".

SAVe:SESsion (No Query Form)

Saves the state of the instrument, including reference waveforms, to a saved session file.

Group Save and Recall

Syntax `SAVe:SESsion <QString>`

Arguments `<QString>` 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 FILESystem:CWDFILESystem: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:SESSION` "c:/rose_was_here.tss" 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_NORMal|_SV_AVErage|_SV_MAXHold|
_SV_MINHold|_MAG_VS_TIME|_FREQ_VS_TIME| _PHASE_VS_TIME
_SV_BASEBAND_IQ]|MATH<x>|REF<x>|ALL| },<QString>
```

Related Commands FILESystem:CWD

SAVe:WAVEform:SOURCELIst?

Arguments <x> is the number of the analog channel (CH<x>), math (MATH<x>), or reference (REF<x>) 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_NORMal saves the Normal Spectrum view waveform of the specified channel.

_SV_AVErage saves the Average Spectrum view waveform of the specified channel.

_SV_MAXHold saves the Maximum Hold Spectrum view waveform of the specified channel.

_SV_MINHold 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 Freuency 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 agument 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 TEK00000.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

```
SAVe:WAVEform:GATing {NONe|CURSors|SCREEN|RESAMPLE|SELected}
SAVe:WAVEform:GATing?
```

Arguments

NONe saves the full waveform data.

CURSors saves the waveform data located between the vertical cursors.

SCREEN saves the waveform data that is on the screen. Nothing outside the waveform will be saved.

RESAMPLE 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.

SELected saves the data from the currently selected history or FastFrame acquisition.

Examples

SAVE:WAVEFORM:GATING NONe saves the entire waveform.

SAVE:WAVEFORM:GATING? might return :SAVe:WAVEform:GATing SCREEN, indicating the save waveform operation is set to save the waveform data on the screen.

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

```
SAVe:WAVEform:GATing:RESAMPLErate <NR1>
SAVe:WAVEform:GATing:RESAMPLErate?
```

Arguments

<NR1> specifies the resample interval.

Examples

SAVE:WAVEFORM:GATING:RESAMPLERATE 3 sets to save waveform data every third data point.

SAVE:WAVEFORM:GATING:RESAMPLERATE? might return :SAVe:WAVEform:GATing:RESAMPLErate 2, 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.

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:TRIGer 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:TRIGer 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

```
SAVEON:WAVEform {<NR1>|ON|OFF}
SAVEON:WAVEform?
```

Related Commands

SAVEON:FILE:DEST

SAVEON:FILE:NAME

SAVEON:TRIGger

Arguments

<NR1> = 0 disables Save On Waveform; any other value turns this feature on.

OFF disables Save On Waveform.

ON enables Save On Waveform.

Examples

SAVEON:WAVEFORM ON turns on the Save On Waveform feature, so that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEFORM? might return :SAVEON:WAVEFORM ON, 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

```
SAVEON:WAVEform:FILEFormat {INTERNal|SPREADSheet}
SAVEON:WAVEform:FILEFormat?
```

Arguments

INTERNal specifies saving the waveform in the instrument internal format.

SPREADSheet specifies saving the waveform in comma separated values format.

Examples

SAVEON:WAVEFORM:FILEFORMAT SPREADSheet 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

`CH<x>` specifies an analog channel to use as the save on source.

`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).

`MATH<x>` specifies a math waveform to use as the save on source.

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

`ALL` specifies all analog, math, digital, and reference waveforms as the source waveforms for saving.

Examples

`SAVEON:WAVEform:SOURce MATH1` specifies MATH 1 as the save on source.

`SAVEON:WAVEform:SOURce?` might return `:SAVEON:WAVEFORM:SOURCE REF1` indicating the save on source is REF1.

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 <Qstring>
SAVEONEVent:FILEDest?
```

Arguments

`<QString>` specifies the location to store files.

Examples

`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.

`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.

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

`SAVEONEVENT:FILENAME "MaskFailure"` sets the name of the file to MaskFailure.

`SAVEONEVENT:FILENAME?` might return `:SAVEONEVent:FILEName "MaskFailure5"`, indicating the name you set for the instrument to use. The autoincrement number of 5 is appended.

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

`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

`SAVEONEVENT:IMAGE:FILEFORMAT JPG` sets the image file format to JPEG.

`SAVEONEVENT:IMAGE:FILEFORMAT?` might return `:SAVEON:IMAGE:FILEFORMAT PNG` indicating that the file format is set to PNG.

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

```
SAVEONEVent:WAVEform:FILEFormat
{INTERNal|SPREADSheet|MATlab}
SAVEONEVent:WAVEform:FILEFormat?
```

Arguments

`INTERNal` specifies saving the waveform in the instrument internal format.

`SPREADSheet` specifies saving the waveform in comma separated values format.

`MATlab` specifies saving the waveform in matlab compatible file format.

Examples

`SAVEONEVENT:WAVEFORM:FILEFORMAT SPREADSheet` sets the file format to spreadsheet (csv).

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

CH<x> specifies an analog channel as the source waveform for saving.

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).

Math<x> specifies a math waveform as the source waveform for saving.

REF<x> specifies a reference waveform as the source waveform for saving.

ALL specifies all analog, math, digital, and reference waveforms as the source waveforms for saving.

Examples

SAVEONEVENT:WAVEFORM:SOURCE MATH1 specifies MATH 1 as the save on source.

SAVEONEVENT:WAVEFORM:SOURCE? might return :SAVEONEVent:IMAGe:FILEFormat REF1, indicating the save on source is REF 1.

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 `NEXT` goes to the next search mark.

`PREVious` goes to the previous search mark.

`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).

`NONE` is the default setting when no action is being taken.

`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).

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 following CAN bus trigger conditions:
`SOF|FRAMEtype|IDentifier|DATa|IDANDDATA|EOF|ERRor|FDBITS`

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|MOREEQua|UNEQual|LESSthan|MOREthan}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier?
```

Arguments

Arguments are the following data qualifier types:
`EQUal|LESSEQual|MOREEQua|UNEQual|LESSthan|MOREthan`

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

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue?
```

Arguments

`<QString>`

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue "1111"` sets the data value to 1111.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:VALUE "1010"` indicating the data value is 1010.

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

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType
{ACKMISS|BITSTUFFing|FORMERRor |ANYERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType?
```

Arguments

`ACKMISS` specifies a search based on a missing ACK field.

`BITSTUFFing` specifies a search based on a bit stuffing error.

`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.

`ANYERRor` specifies a search based on any error type.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE ACKMISS` specifies searching for any missing ACK fields.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE?` might return ANYERROR, indicating that the bus is being searched for all error types.

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 following available frame types: DATa|ERRor|OVERLoad|REMote.

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 following available trigger conditions:
`DATA|IDANDDATA|ERRor|IDentifier|SLEEP|SYNCfield|WAKEup`

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`

"XXXX

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 "XXXXXXXXXXXXXXX10010", indicating that the data value string is set to "XXXXXXXXXXXXXXX10010" 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 "XXXXXXX1", indicating that the data string value for RS232C bus trigger search 1 is set to "XXXXXXX1" 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 XXXXXXXXXXXX, 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 {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|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.

`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: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 <NR1>
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|MOREEQualINrange|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 SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 0.0E+0 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
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold?
```

Arguments <NR3> is the logic clock threshold.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:CLOCk:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:CLOCk:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:CLOCK:THRESHOLD 0.0E+0 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
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime?
```

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

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.

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

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce CH2 sets the clock source to channel 2.

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.

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

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>?
```

Arguments

<NR3> is the voltage level to use for logic trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:CH2 50.0e-3 sets the level to 50 mV.

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.

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:LEVel:MATH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>?
```

Arguments

<NR3> is the voltage level to use for logic trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:MATH1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel: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:LEVel: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:LEVel:REF<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>?
```

Arguments

<NR3> is the voltage level to use for logic trigger search.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:REF1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel: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

OFF specifies not to use the clock source.

ON specifies to use the clock source.

0 specifies not to use the clock source.

1 specifies to use the clock source.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:USEClockedge ON specifies to use the clock source.

SEARCH:SEARCH1:TRIGger:A:LOGIc:USEClockedge? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:USECLOCKEDGE 0 indicating not to use the clock source.

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

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn
{TRUe|FALSe|MOREThan|LESSThan|EQual|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn?
```

Arguments

TRUe searches on an input value that is true.

FALSe searches on an input value that is false.

MOREthan searches on an input value that is greater than a set value.

LESSthan searches on an input value that is less than a set value.

EQual searches on an input value that is equal to a set value.

UNEQual searches on an input value that is not equal to a set value.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:WHEn FALSE specifies a search on an input value that is false.

SEARCH:SEARCH1:TRIGger:A:LOGIc:WHEn? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:WHEN TRUEindicating a search on an input value that is true.

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>| MATH<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 ±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 ±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 following available sources: CH<x>|REF<x>

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 thresholdo 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|EQual}NOTEQual}
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.

`EQual` 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 ±5% tolerance.

NOTEQual argument sets the instrument to search when the pattern is true for atime period greater than or less than (but not equal) the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth within a ±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> isi 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}
SEARCH:SEARCH<x>:TRIGger:A:STATE?
```

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 beow 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
SEARCH:SELected?
```

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.

SECurity:LOGger:STATe

This command sets or queries the state of the security log. The security log records security activities to the securitylog.txt file, which is present when exporting log files. When OFF, the security log stops recording new activity, but all previous activity is still present when exporting log files.

Group Miscellaneous

Syntax
```
SECurity:LOGger:STATe {ON|OFF|<NR1>}
SECurity:LOGger:STATe?
```

Arguments ON 0 disables the security logger; any other value enables the security logger

OFF disables the security logger.

<NR1> turns on the display of the specified waveform. This waveform also becomes the selected waveform.

Examples SECurity:LOGger:STATe ON enables the security logger.

SECurity:LOGger:STATe? might return :SECurity:LOGger:STATe 0 indicating the security logger is disabled.

SELect:CH<x>

Turns the display of the channel <x> waveform on or off, where <x > is the channel number. This command also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the specified waveform.

Group Vertical

Syntax
```
SELect:CH<x> {ON|OFF|1|0}
SELect:CH<x>?
```

Arguments ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

1 turns on the display of the specified waveform. This waveform also becomes the selected waveform.

0 turns off the display of the specified waveform.

Examples SELECT:CH2 ON turns the channel 2 waveform display on, and selects channel 2.

SELECT:CH1? might return :SELECT:CH1 1 indicating that channel 1 is being displayed.

SELect:DCH<x>:DAll

This command turns on or off all constituent digital channels.

Group Digital

Syntax

```
SELect:DCH<x>:DAll {ON|OFF|<NR1>}
SELect:DCH<x>:DAll?
```

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:DAll ON turns ON all the bits of separate digital channel 1.

SELect:DCH1:DAll? might return :SELect:DCH1:DAll 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;:CMDBATCH 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 {TERMinal|NONe}
SOCKETServer:PROTOCol?
```

Arguments TERMinal specifies terminal protocol for the socket server. When set to TERMinal, 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 hours.

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

`<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).

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

`TRIGGER:A:BUS:B1:CAN:DATA:OFFSET 5` sets the CAN data offset to 5 bytes.

`TRIGGER:A:BUS:B1: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 trigger 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 trigger data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

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

```
TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual}
TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier?
```

Arguments

`LESSthan` sets the instrument to trigger when the data is less than the qualifier value.

`MOREthan` sets the instrument to trigger when the data is greater than the qualifier value.

`EQual` sets the instrument to trigger when the data is equal to the qualifier value.

`UNEQual` sets the instrument to trigger when the data is not equal to the qualifier value.

`LESSEQual` sets the instrument to trigger when the data is less than or equal to the qualifier value.

`MOREEQual` sets the instrument to trigger when the data is greater than or equal to the qualifier value.

Examples

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN` sets the instrument to trigger when the data is less than the qualifier value.

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER?` might return `:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN`, indicating that the instrument is set to trigger when the data is less than the qualifier value.

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

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

TRIGGER:A:BUS:B1:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1.

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.

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

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

TRIGGER:A:BUS:B1:CAN:FD:ESIBit ONE specifies filtering CAN FD packets for those where the ESI bit has a value of 1.

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.

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 "XXXXXXXXXXX"` indicating the identifier values is XXXXXXXXXXX.

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 I²C address mode.

`ADDR10` specifies the 10-bit I²C 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 I²C 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 XXXXXXX.

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|IDentifier|DATa|IDANDDATA|WAKEup|SLEEP|ERRor}

TRIGger:A:BUS:B<x>:LIN:CONDition?
```

Arguments SYNCfield sets the LIN trigger condition to sync field.

IDentifier 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|EQual|UNEQual|LESSEQual|MOREEQual|
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.

`EQual` sets the LIN data qualifier to equal.

`UNEQual` sets the LIN data qualifier to not equal.

`LESSEQual` sets the LIN data qualifier to less than or equal.

`MOREEQual` 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

SYNC sets the LIN error type to SYNC.

PARity sets the LIN error type to parity.

CHecksum sets the LIN error type to checksum.

Examples

TRIGGER:A:BUS:B1:LIN:ERRTYPE CHECKSUM sets the LIN error type to checksum.

TRIGGER:A:BUS:B1:LIN:ERRTYPE? might return TRIGGER:A:BUS:B1:LIN:ERRTYPE SYNC indicating the LIN error type is SYNC.

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

TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "110010" sets the identifier value to 110010.

TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE? might return TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "XXXXXX" indicating the identifier value is XXXXXX.

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.

B<x> is the specified bus number.

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 <Qstring> sets the Fast Channel 1 binary data high value.

Examples TRIGger:A:BUS:B12:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX" sets the SENT bus B12 Fast Channel 1 high value to XXXXXXXXXXXX, 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.

B<x> is the specified bus number.

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:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands TRIGger:A:BUS:B<x>:SENT:CONDition

TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:VALue

Arguments `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:CHAN1A:QUALIFIER UNEQUAL` sets the fast channel 1 data qualifier trigger to not equal for bus 2.

`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.

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.

B<x> is the specified bus number.

Conditions Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

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

<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 XXXXXXXXXXXX, 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.

B<x> is the specified bus number.

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

<Qstring> sets the Fast Channel 2 high binary data value.

Examples

TRIGger:A:BUS:B2:SENT:FAST:CHAN2B:HIVALue "100000000000" sets the SENT bus B2 Fast Channel 2 high value to 100000000000.

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.

B<x> is the specified bus number.

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 `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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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

`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: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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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

`<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.

B<x> is the specified bus number.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to PAUSE.

Group Trigger

Syntax

```
TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier {EQual|UNEQual|
LESSthan|MOREthan|LESSEQual|MOREEQual|Inrange|OUTrange}
TRIGger:A:BUS:B<x>:SENT:PAUSE:QUALifier?
```

Related Commands TRIGger:A:BUS:B<x>:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

Arguments `EQUal` sets the qualifier as Equal.

`INrange` sets the qualifier to in 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 out of range.

`UNEQual` sets the qualifier as Unequal.

Examples `TRIGGER:A:BUS:B2:SENT:PAUSE:QUALifier UNEQUAL` sets the pause pulse qualifier trigger to not equal for bus 2.

`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.

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.

B<x> is the specified bus number.

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:HIVALue <QString>
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:HIVALue?
```

Related Commands TRIGger:A:BUS:B<x>:SENT:CONDition

Arguments <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.

B<x> is the specified bus number.

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| INRange| OUTRange}
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:QUALifier?
```

Related Commands TRIGger:A:BUS:B<x>:SENT:CONDition

Arguments 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.

B<x> is the specified bus number.

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

<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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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 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.

B<x> is the specified bus number.

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 <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.

B<x> is the specified bus number.

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 <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>| INTernal|
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.

INTernal 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|<NR1>}
TRIGger:A:LOGIc:USECLockedge?
```

Arguments

`ON|1` specifies that logic trigger type uses clock source.

`OFF|0` 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|EQual|
UNEQual}
TRIGger:A:LOGIc:WHEn?
```

Arguments

TRUe triggers on an input pattern that is true.

FALSe triggers on an input pattern that is false.

MOREthan triggers on an input pattern that is true for a time period greater than a user defined Time Limit (DELTatime) value.

LESSthan triggers on an input pattern that is true for a time period less than a user defined Time Limit (DELTatime) value.

EQual triggers on an input pattern that is true for a time period equal to a user defined Time Limit (DELTatime) value.

UNEQual triggers on an input pattern that is true for a time period not equal to a user defined Time Limit (DELTatime) value.

Examples

TRIGger:A:LOGIc:WHEn EQual specifies triggering when the input pattern is true for a time period equal to a user defined Time Limit value.

TRIGger:A:LOGIc:WHEn? might return :TRIGGER:A:LOGIC:WHEN TRUE indicating a trigger when the logic is true.

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

```
TRIGger:A:PULSEWidth:WHEn
{LESSthan|MOREthan|EQual|UNEQual|WIThin|OUTside}
TRIGger:A:PULSEWidth:WHEn?
```

Related Commands TRIGger:A:PULSEWidth:LOWLimit

TRIGger:A:PULSEWidth:HIGHLimit

TRIGger:A:PULSEWidth:SOUrce

Arguments `LESSthan` causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:A:PULSEWidth:LOWLimit command.

`MOREthan` causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:A:PULSEWidth:LOWLimit command.

`EQual` causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:A:PULSEWidth:LOWLimit within a ±5% tolerance.

`UNEQual` 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 ±5% tolerance.

`WIThin` causes a trigger when a pulse is detected that is within a range set by two values.

`OUTside` causes a trigger when a pulse is detected that is outside of a range set by two values.

Examples `: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.

`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.

`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.

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 `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.

`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.

`EITher` triggers on a runt of either polarity.

Examples `TRIGger:A:RUNT:POLarity NEGATIVE` specifies that the polarity of the A pulse runt trigger is negative.

`TRIGger:A:RUNT:POLarity?` might return `:TRIGGER:A:RUNT:POLARITY POSITIVE` indicating that the polarity of the A pulse runt trigger is positive.

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 `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>.

Examples `TRIGger:A:RUNT:SOUrce CH4` sets channel 4 as the trigger source.

`TRIGger:A:RUNT:SOUrce?` might return `:TRIGGER:A:RUNT:SOURCE CH2` indicating that channel 2 is the trigger source.

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|EQual|UNEQual|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.

`EQual` 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 ±5% tolerance.

`UNEQual` 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 ±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 µs.

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 CH<x> specifies the analog channel to use as the clock source waveform.

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.

Examples TRIGger:A:SETHold:CLOCk:SOUrce CH1 specifies channel 1 as the clock source for a setup and hold trigger operation.

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.

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
```
TRIGger:A:SETHold:HOLDTime <NR3>
TRIGger:A:SETHold:HOLDTime?
```

Arguments <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.

Examples TRIGger:A:SETHold:HOLDTime 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.

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.

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 µs.

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:TRANsition:DELTatime

This command specifies the delta time (that is Time Limit) used in calculating the transition value for the transition (that is Rise or Fall Time) trigger.

Group Trigger

Syntax
```
TRIGger:A:TRANsition:DELTatime <NR3>
TRIGger:A:TRANsition:DELTatime?
```

Arguments <NR3> is a floating point number that specifies the delta time, in seconds.

Examples TRIGGER:A:TRANSITION:DELTATIME 15E-6 sets the delta time of the transition trigger to 15 µs.

TRIGGER:A:TRANSITION:DELTATIME? might return TRIGGER:A:TRANSITION:DELTATIME 2.0000E-09, indicating that the delta time of the transition trigger is set to 2 ns.

TRIGger:A:TRANsition:LOGICQUALification

This command specifies whether or not to use logic qualification for a transition trigger.

Group Trigger

Syntax
```
TRIGger:A:TRANsition:LOGICQUALification {ON|OFF}
TRIGger:A:TRANsition:LOGICQUALification?
```

Arguments ON specifies that the transition trigger type uses logic qualification.

OFF specifies that the transition trigger type does not use logic qualification.

Examples TRIGGER:A:TRANsition:LOGICQUALification ON specifies using logic qualification.

TRIGGER:A:TRANsition:LOGICQUALification? might return :TRIGGER:A:TRANsition:LOGICQUALification OFF indicating logic qualification is off.

TRIGger:A:TRANsition:POLarity

This command specifies the polarity for the transition trigger.

Group Trigger

Syntax

```
TRIGger:A:TRANsition:POLarity {EITher|NEGative|POSitive}
TRIGger:A:TRANsition:POLarity?
```

Arguments

`EITher` indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

`NEGative` indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

`POSitive` indicates either positive or negative polarity.

Examples

`TRIGGER:A:TRANSITION:POLARITY NEGATIVE` sets the transition polarity to negative.

`TRIGGER:A:TRANSITION:POLARITY?` might return `:TRIGGER:A:TRANSITION:POLARITY EITHER`, indicating that the polarity can be either positive or negative.

TRIGger:A:TRANsition:SOUrce

This command specifies the source waveform for a transition trigger. Digital channels are not supported as transition trigger sources.

Group

Trigger

Syntax

```
TRIGger:A:TRANsition:SOUrce {CH<x>}
TRIGger:A:TRANsition:SOUrce?
```

Arguments

`CH<x>` specifies one of the analog channels to be used as the source for a transition trigger.

Examples

`TRIGGER:A:TRANSITION:SOURCE CH4` sets channel 4 as the source for the transition trigger.

`TRIGGER:A:TRANSITION:SOURCE CH4?` might return `:TRIGGER:A:TRANSITION:SOURCE CH2`, indicating that channel 2 is the source for the transition trigger.

TRIGger:A:TRANsition:WHEn

his command specifies whether to check for a transitioning signal that is faster or slower than the specified delta time.

Group Trigger

Syntax
```
TRIGger:A:TRANsition:WHEn {SLOWer|FASTer|EQual|UNEQual}
TRIGger:A:TRANsition:WHEn?
```

Arguments

`SLOWer` sets the trigger to occur when the signal transition time is slower than the time set by.TRIGger:A:TRANsition:DELTatime.

`FASTer` sets the trigger to occur when the signal transition time is faster than the time set by TRIGger:A:TRANsition:DELTatime.

`EQual` sets the trigger to occur when the signal transition time is equal to the time set by TRIGger:A:TRANsition:DELTatime.

`UNEQual` sets the trigger to occur when the signal transition time is not equal to the time set by TRIGger:A:TRANsition:DELTatime.

Examples

`TRIGGER:A:TRANSITION:WHEN SLOWER` sets the trigger to occur when the signal transition time is slower than the time set by the TRIGger:A:TRANsition:DELTatime command.

`TRIGGER:A:TRANSITION:WHEN?` might return `:TRIGGER:A:TRANSITION:WHEN FASTER`, indicating that the trigger will occur when the signal transition time is faster than the time set by the TRIGger:A:TRANsition:DELTatime command.

TRIGger:A:TYPe

This command sets or queries the type of A trigger.

Group Trigger

Syntax
```
TRIGger:A:TYPe {EDGE| WIDth| TRANsition| TIMEOut| RUNt|
LOGIc| SETHold| 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.

`TRANsition` specifies that a trigger occurs when a specified pulse is found that meets the transition trigger parameters.

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.

SETHold 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

TRIGGER:A:UPPERTHRESHOLD:CH1 1.3 This command sets the A trigger level for Channel 1 to 1.3 V level.

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.

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

```
TRIGger:A:HOLDoff:TIMe <NR3>
TRIGger:A:HOLDoff:TIMe?
```

Arguments <NR3> specifies the holdoff time in seconds. The range is from 0 seconds through 10 seconds.

Examples TRIGGER:A:HOLDOFF:TIME 10 sets the A trigger holdoff time to 10 s.

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.

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

```
TRIGger:A:MODe {AUTO|NORMal}
TRIGger:A:MODe?
```

Arguments AUTO generates a trigger if one is not detected within a specified time period.

NORMal waits for a valid trigger event.

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

```
VERTical:DESKew:FROM:CUSTOMPROPAgation <NR3>
VERTical:DESKew:FROM:CUSTOMPROPAgation?
```

Arguments <NR3> 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 VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.25e-9 sets the custom delay to 250 ps.

VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION? might return :VERTICAL:DESKew:FROM:CUSTOMPROPAGATION 0.0E+0 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

```
VERTical:DESKew:FROM:SOUrce CH<x>
VERTical:DESKew:FROM:SOUrce?
```

Arguments The vertical deskew source CH<x>.

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 CH<x>.

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.

VXI:ENAble

This command sets or queries the state of the VXI-11 server, which is used for command and control over an Ethernet connection.

Group Miscellaneous

Syntax

```
VXI:ENAble {ON|OFF|1|0}
VXI:ENAble?
```

Arguments `ON` enables the VXI-11 server.

`OFF` disables the VXI-11 server.

`1` enables the VXI-11 server.

`0` disables the VXI-11 server.

Examples `VXI:ENAble ON` enables the VXI-11 server.

`VXI:ENAble?` might return `:VXI:ENAble OFF`, indicating the VXI-11 server is disabled.

VXI:PORT:HIGH

This command sets or queries the higher end of the port range for the VXI-11 server.

Group Miscellaneous

Syntax

```
VXI:PORT:HIGH <NR1>
VXI:PORT:HIGH?
```

Arguments <NR1> is the high end of the TCPIP port range for the VXI-11 server connection. Some ports are restricted and cannot be used because they are used by other services. If a restricted port falls between the VXI-11 server low port and the VXI-11 server high port, then the error code 224 (Illegal parameter value) is posted to the event queue and the VXI-11 high port remains unchanged. If the VXI-11 high port number is set to a valid value below the VXI-11 low port number, the VXI-11 low port number is set to the value of the VXI-11 high port number minus one.

Examples VXI:PORT:HIGH 7010 sets the VXI-11 high port number to 7010.

VXI:PORT:HIGH? might return :VXI:PORT:HIGH 7010, indicating that the port number is 7010.

VXI:PORT:LOW

This command sets or queries the lower end of the port range for the VXI-11 server.

Group Miscellaneous

Syntax

```
VXI:PORT:LOW <NR1>
VXI:PORT:LOW?
```

Arguments <NR1> is the low end of the TCPIP port range for the VXI-11 server connection. Some ports are restricted and cannot be used because they are used by other services. If a restricted port falls between the VXI-11 server low port and the VXI-11 server high port, then the error code 224 (Illegal parameter value) is posted to the event queue and the VXI-11 low port remains unchanged. If the VXI-11 low port number is set to a valid value above the VXI-11 high port number, the VXI-11 high port number is set to the value of the VXI-11 low port number plus one.

Examples

`VXI:PORT:LOW 7000` sets the VXI-11 low port number to 7010.

`VXI:PORT:LOW?` might return `:VXI:PORT:LOW 7000`, indicating that the port number is 7000.

*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

WFMOutpre?

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,Sampl mode".

WFMOutpre? (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 WFMOutpre?

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 `WFMOUTPRE:DOMAIN?` might return `WFMOUTPRE:DOMAIN TIME`, indicating that the outgoing waveform is a time domain trace.

`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).

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 `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.

`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.

Examples `WFMOUTPRE:ENCDG ASCii` specifies that the outgoing waveform data will be sent in ASCII format.

`WFMOUTPRE:ENCDG?` might return `WFMOUTPRE:ENCDG BINary`, indicating that outgoing waveform data will be sent in binary format.

WFMOutpre: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
```
WFMOutpre:NR_Pt?
```

Related Commands CURVe?

DATa

DATa:STARt

DATa:STOP

SAVe:WAVEform

SAVEON:WAVEform:FILEFormat

Examples `WFMOUTPRE:NR_PT?` might return `WFMOUTPRE:NR_PT 5000`, indicating that there are 5000 data points to be sent.

WFMOutpre: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 `WFMOutpre: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 251 (vertical) by 1024 (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 1024 possible horizontal columns being transmitted contains 251 vertical points. Note: 251 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 forth 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"
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 µVs/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"

Table 2-48: Waveform Suffixes (cont.)

Field	Description	Examples
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,500.0µs/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_AVErage, 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 `WFMOUTPRE:XINCR?` might return `WFMOUTPRE:XINCR 10.0000E-6`, indicating that the horizontal sampling interval is 10 µs/point (500 µs/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:YMUlt? (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:YMUlt? 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.*

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

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.

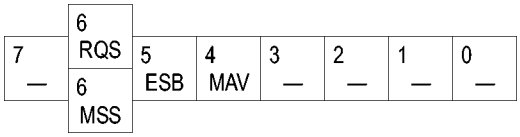


Figure 3-2: The Status Byte Register (SBR)

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.

Table 3-2: SBR Bit Functions (cont.)

Bit	Function	
4	MAV	`Message Available.` Shows that output is available in the Output Queue.
3	————	Not used.
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.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

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.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

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.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

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.

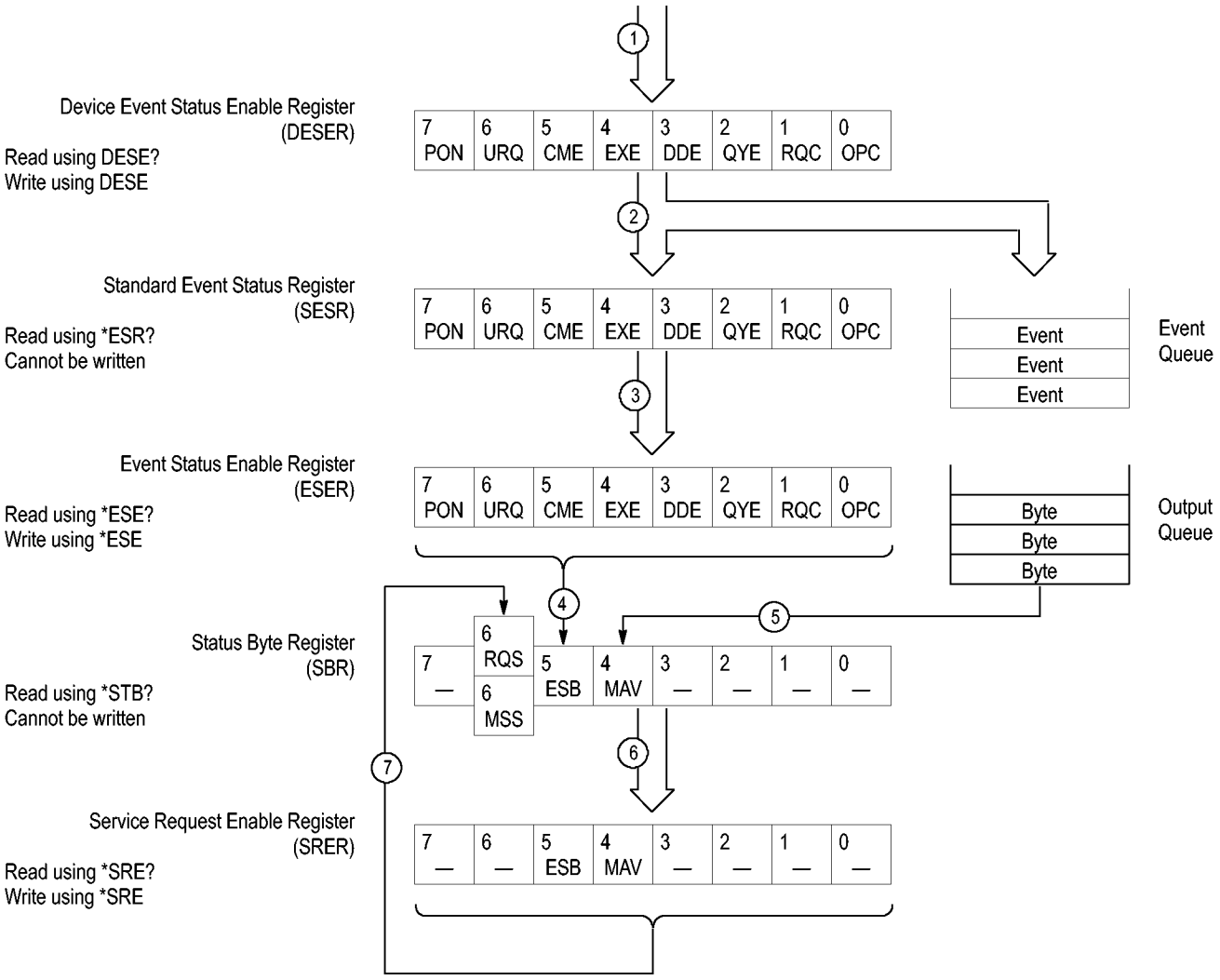


Figure 3-6: Status and Event Handling Process

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 ESER 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 Regiser) 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:INTERNal	
CALibrate:INTERNal:STARt	
CALibrate:FACtory STARt	
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.

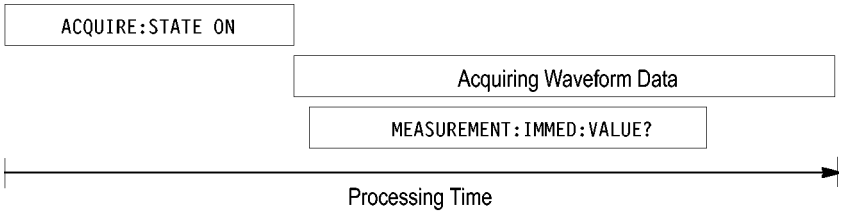


Figure 3-7: Command Processing Without Using Synchronization

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?.

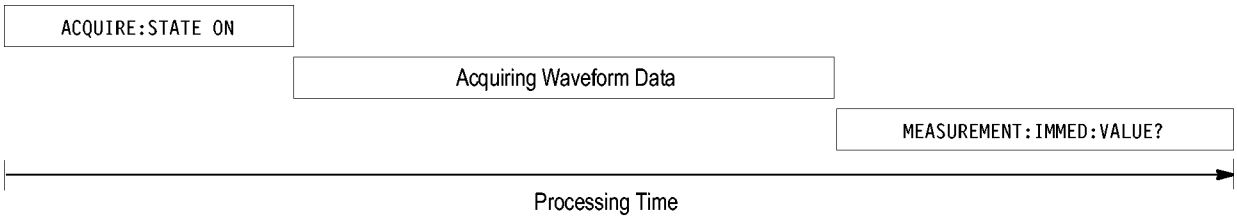


Figure 3-8: Processing Sequence With Synchronization

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
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
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

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

Code	Message
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
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

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

Code	Message
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
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.

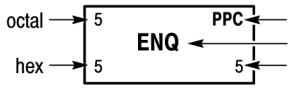
Table 3-11: Internal Warning Messages

Code	Message
630	Internal warning, 50Ω overload

Appendix A: Character Set

B7 B6 B5 / BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 LA0 SP 20 32	60 LA16 0 30 48	100 TA0 @ 40 64	120 TA16 P 50 80	140 SA0 ` 60 96	160 SA16 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LL0 DC1 11 17	41 LA1 ! 21 33	61 LA17 1 31 49	101 TA1 A 41 65	121 TA17 Q 51 81	141 SA1 a 61 97	161 SA17 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 LA2 " 22 34	62 LA18 2 32 50	102 TA2 B 42 66	122 TA18 R 52 82	142 SA2 b 62 98	162 SA18 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 LA3 # 23 35	63 LA19 3 33 51	103 TA3 C 43 67	123 TA19 S 53 83	143 SA3 c 63 99	163 SA19 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 LA4 $ 24 36	64 LA20 4 34 52	104 TA4 D 44 68	124 TA20 T 54 84	144 SA4 d 64 100	164 SA20 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 LA5 % 25 37	65 LA21 5 35 53	105 TA5 E 45 69	125 TA21 U 55 85	145 SA5 e 65 101	165 SA21 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 LA6 & 26 38	66 LA22 6 36 54	106 TA6 F 46 70	126 TA22 V 56 86	146 SA6 f 66 102	166 SA22 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 LA7 ' 27 39	67 LA23 7 37 55	107 TA7 G 47 71	127 TA23 W 57 87	147 SA7 g 67 103	167 SA23 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 LA8 (28 40	70 LA24 8 38 56	110 TA8 H 48 72	130 TA24 X 58 88	150 SA8 h 68 104	170 SA24 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51 LA9) 29 41	71 LA25 9 39 57	111 TA9 I 49 73	131 TA25 Y 59 89	151 SA9 i 69 105	171 SA25 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 LA10 * 2A 42	72 LA26 : 3A 58	112 TA10 J 4A 74	132 TA26 Z 5A 90	152 SA10 j 6A 106	172 SA26 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 LA11 + 2B 43	73 LA27 ; 3B 59	113 TA11 K 4B 75	133 TA27 [5B 91	153 SA11 k 6B 107	173 SA27 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 LA12 , 2C 44	74 LA28 < 3C 60	114 TA12 L 4C 76	134 TA28 \ 5C 92	154 SA12 l 6C 108	174 SA28 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 LA13 – 2D 45	75 LA29 = 3D 61	115 TA13 M 4D 77	135 TA29] 5D 93	155 SA13 m 6D 109	175 SA29 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 LA14 . 2E 46	76 LA30 > 3E 62	116 TA14 N 4E 78	136 TA30 ^ 5E 94	156 SA14 n 6E 110	176 SA30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 LA15 / 2F 47	77 UNL ? 3F 63	117 TA15 O 4F 79	137 UNT – 5F 95	157 SA15 o 6F 111	177 RUBOUT (DEL) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY



Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

*CAL
*CLS
*DDT
*ESE
*ESR
*IDN
*LRN
*OPC
*OPT
*PSC
*PUD
*RST
*SRE
*STB
*TRG
*TST
*WAI
0
1
1066
1200
128
1333
1466
15
1600
1866
2133
333
500E1
500E15
500E3
7
8
800
9
A
AB
ABC
ABCB
ABORt
ABOrt
ABSOLUTE
ABSolute
AC
ACBC
ACCM
ACCOMMONMODE
ACDC
ACDCRMS
ACK
ACKMISS
ACKNowledge
ACKnum
ACQ
ACQDURATION
ACQuire
ACQuisition
ACRMS
ACTION
ACTONEVent
ACTive
ADD
ADDMEAS
ADDNew
ADDR10
ADDR7
ADDRANDDATA
ADDRESS
ADDRess
ADDress
ADVanced
AFG
AGND
AHPOSition
ALARMSEARch
ALERT
ALIas
ALL
ALLAcqs
ALLBits
ALLEv
ALLTHResholds
ALWAYS
AM14
AMPL
AMPLINear
AMPLITUDE
AMPLitude
AMPMode
AMP[1-10]Val
AMPlITUDE
AN
ANALOG
ANALOg
ANALYSISMethod
ANALYZemode
AND
ANNOTate
ANOISEDensity
ANY
ANYERROr
AOFFSet
APOSition
APPID
APPPWR
APPPWRSUM
APPly
ARBitrary
AREA
AREAlt;x>
AREAofcrosssection
ARROW
ASCII
ASCIi
ASC_Fmt
ASCii
ASIC
ASOUrce
ASPEctratio
ATRIGger
AUDio
AUNIts
AUTHenticate
AUTO
AUTODim
AUTOETHERnet
AUTOINCrement
AUTORange
AUTORbw
AUTOSAVEPITIMEOUT
AUTOSAVEUITIMEOUT
AUTOScale
AUTOSet
AUTOZero
AUTOfit
AUTOmatic
AUTOset
AUTOset
DPMAutoset
AUTOset DPMPReset
AUXLevel
AUXiliary
AUXout
AVAILable
AVErage
AVG
AVPOSition
AXPOSition
AYPOSition
Auto
B
B0
B1
B10
B11
B12
B13
B14
B15
B16
B2
B3
B4
B5
B6
B7
B8
B9
BACA
BACKLight
BACKWard
BADDr
BADGE
BADGe
BANDWidth
BANdwidth
BASE
BASETop
BASic
BATHTUB
BATHtub
BC
BCR

BCRType
BDIFFBP
BEACon
BER
BHPOSition
BIAS
BIN
BINARY
BINary
BIT
BITAMPLITUDE
BITCfgmode
BITDelay
BITEnd
BITHIGH
BITLOW
BITORDer
BITOrder
BITPERiod
BITPcnt
BITRate
BITS
BITSTUFFing
BITSTart
BITType
BIT_Nr
BITlt;n>SOUrce
BJT
BLACKMANHarris
BLACkmanharris
BLANKingtime
BLOCKDATa
BLOCkid
BLUe
BM
BMP
BMSGEnd
BN
BNOISE
BN_Fmt
BOFFSet
BOLD
BOOKMARK
BOOKMark
BOTH
BPACKets
BPOSition
BRGTID
BRINgtoview
BROADCASTPacket
BROadcast
BRSBit
BSOUrce
BULK
BUNIts
BURSTDETectmethod
BURSTEDGTYPe
BURSTLENGTH
BURSTLatency
BURSTWIDTH
BURSt
BUS
BUSANDWAVEFORM
BUSTABle
BUSTURNAROUnd
BUSY
BUSx
BVPOSition
BXPOSition
BY
BYPOSition
BYT_Nr
BYT_Or
BYTe
Block
Blt;x>
Blt;x>PSIFIVe
Bpeak
Br
Bus
Bx
C
CA
CALLOUTS
CALLOUTlt;x>
CALibrate
CAN
CAN2X
CANH
CANL
CAPACITANCE
CARDIac
CATalog
CB
CCODe
CCOUnt
CCRESUlts
CCYCles
CENTERFREQuency
CENTERFrequency
CGND
CGS
CH
CH1
CH2
CH3
CH4
CH5
CH6
CH7
CH8
CHAN1A
CHAN2B
CHANWidth
CHX
CHannel
CHecksum
CHlt;x>
CHlt;x>_DALL
CHlt;x>_Dlt;x>
CHx
CHx_DALL
CHx_Dx
CHx_Dy
CLASSA
CLASSB
CLASSC
CLASSD
CLASs
CLEAR
CLEar
CLIPping
CLOCK
CLOCKBITSPERCHANNEL
CLOCKFrequency
CLOCKMultiplier
CLOCKRecovery
CLOCKSOUrce
CLOCk
CLOCkSOUrce
CLRESPONSE
CMEThod
CN
COFFSet
COLOR
COLOROFF
COLORON
COLORy
COLor
COLors
COMM
COMMAND
COMMONMODE
COMMONMode
COMMand
COMMents
COMPRESSEDPSTReam
COMPRESSION
COMPensate
COMPlete
COMPosition
COMPression
CONDENRG
CONDLOSS
CONDition
CONFIGuration
CONFIGure
CONFigure
CONNECTSTATus
CONNect
CONNected
CONSTAMPlitude
CONSTANTCLOCK
CONSTCLOCKMODe
CONSTant
CONTROL
CONTROLCHARTYPe
CONTROLCODETYPe
CONTROLMESSage
CONTRol
CONTinuous
CONTrol
COPy
COUNT
COUNTer
COUNt
COUPling
COUnt
CPHY
CR
CRC
CRC16
CRC5
CRCERRor
CRCHeader
CRCTrailer
CREATor
CROSSIng
CSActive
CSI
CSIBLANk
CSINULL
CSLevel
CSMode
CSPLIT
CSSource
CSYMVALue
CURRENT
CURRent
CURRentacq
CURSOR
CURSOR1

CURSORlt;x>
CURSor
CURSors
CURVe
CURrent
CUSTOM
CUSTOMLIMITSFile
CUSTOMPROPAgation
CUSTOMTABle
CUSTom
CWD
CYCLE
CYCLEAmp
CYCLEBase
CYCLECount
CYCLEMAX
CYCLEMin
CYCLEMode
CYCLEPKPK
CYCLETop
CYCLEcount
CYCLemode
ChipSelect
Chx_Dx
ClockEdge
Current
D
DADDr
DAMPing
DATA
DATA0
DATA1
DATA2
DATAA
DATAB
DATABITS
DATABITs
DATABits
DATAFORMat
DATAFORmat
DATAFormat
DATAMINUS
DATAMINUSTHRESHOLD
DATAMINUSTHRESHold
DATAPLUS
DATAPLUSTHRESHold
DATAPacket
DATAPath
DATARATE
DATARate
DATASource
DATATHRESHold
DATE
DATa
DBCA
DBM
DBMA
DBMV
DBUA
DBUV
DBUW
DC
DCD
DCDC
DCP1W2
DCPWR
DCR
DCREJ
DCRType
DCSGLONGWRITE
DCSLONGREAD
DCSR
DCSSRR
DCSSW
DDJ
DDR
DDR3
DDRAOS
DDRAOSPERTCK
DDRAOSPERUI
DDRAUS
DDRAUSPERTCK
DDRAUSPERUI
DDRHOLDDIFF
DDRREAD
DDRREADWRITE
DDRREADWrite
DDRRead
DDRSETUPDIFF
DDRTCHABS
DDRTCHAVERAGE
DDRTCKAVERAGE
DDRTCLABS
DDRTCLAVERAGE
DDRTDQSCK
DDRTERRMN
DDRTERRN
DDRTJITCC
DDRTJITDUTY
DDRTJITPER
DDRTPST
DDRTRPRE
DDRTWPRE
DDRVIXAC
DDRWRITE
DDRWrite
DDT
DECIMAL
DECode
DEFect
DEFine
DEFinedby
DEGAUSS
DEGrees
DELAY
DELETE
DELETEALL
DELEte
DELIMiter
DELTA
DELTATime
DELTa
DELTatime
DELay
DELete
DELeteqstring
DELta
DESE
DESKEW
DESKew
DEST
DESTination
DESTinationaddr
DESTinationport
DEVERRor
DEVICEChirp
DEVICEDESCMASTERREAd
DEVICEDESCSLAVEREAd
DEVICETYPe
DEVICEType
DEVicetype
DHCPbootp
DIAg
DIBYDT
DIDT
DIFF
DIFFerential
DIGGRPlt;x>
DIR
DIRECTPacket
DIRacmodel
DIRect
DIRection
DISCMODe
DISCrete
DISLave
DISPLAY
DISPLAYUnits
DISParity
DISPlaystat
DISabled
DISplay
DISplaymode
DIVide
DIVisions
DJ
DJDD
DJDIRAC
DMINus
DMSGEnd
DNS
DOFFSet
DOMAINname
DOMain
DONTInclude
DONTcare
DOTsonly
DOUBLE
DPHY
DPLUs
DPMOVERSHOOT
DPMRIPPLE
DPMTURNOFFTIME
DPMTURNONTIME
DPMUNDERSHOOT
DPlus
DQ0
DQDQS
DRIVE
DRIve
DSI
DSIBLANk
DSINULL
DUty
DVBYDT
DVDT
DVM
DWORd
DYNAMIC
DYNAMic
DYNRange
DeltaB
DeltaH
Dlt;x>
EACHCLOCKCYCLE
EBIT
EBITVALue
ECC
ECL
ECU
ECUDATa
ECUSENSor

ECUSOURce
EDECAy
EDGE
EDGEIncre
EDGEQUALONE
EDGEQUALTWO
EDGEQUALifier
EDGES
EDGESOURce
EDGESource
EDGEType
EDGElt;x>
EDGe
EFFICIENCY
EFFiciency
EIGHTBIT
EIGHTHUNDREdhz
EITHER
EITHer
EIther
EMBEDDED
ENABle
ENAble
ENAbled
ENCdg
ENCoding
END
ENDPoint
ENDSEARch
ENDtime
ENET
ENET100
ENET1000
ENGineering
ENHANCED4
ENHANCED8
ENSLave
ENTERSWindow
ENTRDya
ENTRTSTMode
ENTasx
ENVelope
ENable
EOF
EOFTYPE
EOP
EOPBITS
EOT
EOTP
EOTPDATa
EOW
EOp
EQUal
EQUation
EQual
EQualNOTEQual
ERISe
ERROR
ERRORREPort
ERRORTYPe
ERROr
ERRTYPE
ERRTYPe
ERRType
ERRor
ERRors
ESCAPEMODe
ESIBit
ET
EVEN
EVENT
EVENTBYTe
EVENTS
EVENTtable
EVERY
EVMsg
EVQty
EXCURsion
EXECUTEQUEue
EXECute
EXITSWindow
EXPLICITCLOCK
EXPLICITCLOCKMODe
EXPORTRaw
EXTAtten
EXTDBatten
EXTENDED
EXTENDed
EXTENDuis
EXTREGREAd
EXTREGWRIte
EXTUnits
EXTended
EXTernal
EXTime
EYEDIAGRAM
EYEHIGH
EYELOW
EYEMASK
EYERENDER
EYERender
EYEWIDTHBER
EYEhistogram
Efficiency1
Efficiency2
Efficiency3
Equals
Execute
F1MAG
F2
F3MAG
F4
F8
FACtory
FAIL
FAILCount
FALL
FALLHigh
FALLING
FALLLow
FALLMid
FALLSLEWRATE
FALLTIME
FALLing
FALSe
FALling
FAMILYCODe
FAST
FASTAcq
FASTERthan
FASTer
FASTframe
FAst
FBD1
FBD2
FBD3
FC1063
FC133
FC2125
FC266
FC4250
FC531
FC8500
FCS
FCSError
FD
FDBITS
FDISO
FDNONISO
FFT
FIBREchannel
FIFTY
FIFTYHZ
FIFTy
FIFTyhz
FIFty
FILE
FILEDest
FILEFormat
FILEName
FILESYSTEM
FILESystem
FILTERRANGEFROM
FILTERRANGETO
FILTer
FILTers
FILepath
FIRST
FIVE
FIVEHundred
FIXED
FIXed
FLATNESS
FLATTOP2
FLATtop2
FLAg
FLEXRAY
FLEXray
FLIP
FLYBACK
FOLDer
FONT
FONTSize
FORCEDRange
FORCe
FORMERRor
FORMat
FORWard
FOUR
FOURBIT
FOURHUNDREDHZ
FOURHUNDREdhz
FOURHundred
FOURTEENten
FP
FPAnel
FPBinary
FRAME
FRAMECONTrol
FRAMEEND
FRAMEID
FRAMELENgth
FRAMEREJ
FRAMESTARt
FRAMETYPe
FRAMEType
FRAMEtype
FRAMING
FRAMe
FRAme
FREFerence
FREQ

```
FREQBYTe
FREQUENCY
FREQ[1-10]Val
FREQ[1-11]Val
FREQuency
FROM
FROMEDGESEARCHDIRect
FROMLevel
FROMSearch
FROMSymbol
FROMedge
FRame
FULL
FULLSPEED
FULLScreen
FULl
FUNCTIONCODETYPe
FUNCTIONCODe
FUNCtion
FUNDCURRent
FW1394BS1600B
FW1394BS400B
FW1394BS800B
False
GAIN
GAINCROSSOVERFREQ
GAP
GATESOurce
GATEWay
GATing
GAUSSian
GAUSsian
GENIPADDress
GENerator
GET
GETBusch
GETDevch
GETMRdl
GETMWrl
GETPrid
GETReg
GETRegpktalert
GETRegpktbad
GETRegpktrecent
GETRegtestcfg
GETRegvendor
GETRegvrevent
GETSlave
GLOBal
GLOBalref
GLONGREAD
GLONGWRITE
GLObal
GM
GMT
GPKNOB1
GPKNOB2
GR
GRATICULE
GRATicule
GRAticule
GREATERthan
GREen
GRIDTYPE
GRIDlines
GRId
GROUPDelay
GSLSb
GSMSb
GSP
GSRR
GSW
H
HAMMing
HANDSHAKEPacket
HANDSHAKEType
HANNing
HARMONICS
HARMONICSCLass
HARMONICSSOURce
HARMONICSStd
HARMONICSUNits
HAVERSINe
HBArs
HDRCapability
HDRExit
HDRRestart
HEADER
HEADer
HEIGHT
HEIGHTBER
HEIGht
HEX
HEXAgon
HFRej
HI
HID
HIGH
HIGHLEVel
HIGHLevel
HIGHLimit
HIGHPass
HIGHREFVoltage
HIGHTIME
HIGHZ
HIRes
HISTOGRAM
HIStory
HITDMVALue
HITS
HITType
HIVALue
HOLD
HOLDTime
HOLDoff
HOMEDir
HORDer
HORIZontalscale
HORZ
HORZPOS
HORZScale
HORizontal
HOSTChirp
HOTJoin
HS
HSOURce
HSYNCEND
HSYNCSTART
HUB
HUNDREDBASETX
HUNdred
HYSTeresis
HYSteresis
Hc
Hmax
I2C
I2S
IBA2500
IBA_GEN2
ICFACTOR
ID
IDANDDATA
IDFORmat
IDLE
IDLETime
IDLe
IDentifier
IEC
IEEE519
ILEVELAbs
ILEVELPct
IMAGe
IMAGinary
IMDAACQTREND
IMDADQ0
IMDAEFFICIENCY
IMDAHARMONICS
IMDAINPUTCURRENT
IMDAINPUTPOWER
IMDAINPUTVOLTAGE
IMDALINERIPPLE
IMDAPHASORDIAGRAM
IMDAPLOTDisplay
IMDAPOWERQUALITY
IMDASWITCHRIPPLE
IMDATIMETREND
IMPEDANCE
IMPEDance
IN
INACCBYTe
INCLUDEREFs
INCLUde
INCLude
INDBits
INDEPENDENT
INDUCT
INDUCTANCE
INDex
INFILTers
INFInite
INFMAXimum
INFMINimum
INFPersist
INFormation
INIT
INPUT
INPUTCAP
INPUTLEVel
INPUTMode
INPUTPOwer
INPUTSOurce
INPUTType
INPWR
INPWRSUM
INPut
INRUSH
INRUSHcurrent
INSIDEGreater
INSIDErange
INSTR
INSTall
INTENSITy
INTERNal
INTERPRatio
INTERRUPT
INTERnal
INTERp
INTernal
INVERTColor
INVERTNIBble
INVERTed
INVErted
```

INVert
INVerted
INputs
INrange
IO
IOUT1SOUrce
IOUT2SOUrce
IOUT3SOUrce
IPADDress
IPHeader
IPOWer
IPVFOUR
IRMS
IRipple
ISCLOCKED
ISOALL
ISOEND
ISOMID
ISOSTART
ISOURce
ISOUrce
ISOchronous
ITALIC
ITALic
ITEM
IVSINTEGRALV
IVSINTV
InputPwr
Inrange
J2
J9
JITTERMODE
JITTERSUMMARY
JITTERSummary
JITTermodel
JPG
JTFBandwidth
KAISERBessel
KAISerbessel
L
L2LTOL2N
LABELANDDATA
LABel
LAN
LATCh
LAYout
LDIR
LEFt
LENGth
LENgth
LESSEQual
LESSLimit
LESSThan
LESSthan
LEVELUNIts
LEVel
LF
LFREQuency
LFRej
LIC
LICENSE
LICense
LIMIT
LIMITCount
LIMit
LIN
LINE
LINEAR
LINEAr
LINEEND
LINEFREQUEncy
LINERIPPLE
LINESTARt
LINESelected
LINear
LIST
LJ
LOAd
LOCKCenter
LOCKRJ
LOCKRJValue
LOCKSpectrum
LOCk
LOG
LOGARITHM
LOGIC
LOGIC1SOUrce
LOGIC2SOUrce
LOGIC3SOUrce
LOGIC4SOUrce
LOGIC5SOUrce
LOGICPattern
LOGICQUALification
LOGICSource
LOGIc
LOGic
LONG
LONGEXTREGREAd
LONGEXTREGWRIte
LOOP
LOOPBandwidth
LORENtz
LOW
LOWDATAMINus
LOWDATAPLUS
LOWERFREQuency
LOWLevel
LOWLimit
LOWPass
LOWREFVoltage
LOWSPEED
LOWTHRESHold
LOWTIME
LOWer
LOWerthreshold
LP
LPDDR3
LPDT
LPS666
LPTHRESHold
LPYCBCR24
LRIPPKPK
LRIPRMS
LSB
LSLave
LXI
LogicState
MAC
MACADDRess
MACLENgth
MAG
MAGLOSS
MAGNETICLOSS
MAGNITUDE
MAGNitude
MAGPROPERTY
MAINWindow
MAIn
MANChester
MANUAL
MANual
MARKER
MARgin
MASK
MASKFail
MASKHit
MASKOffset
MASKPass
MASKfile
MASKlt;x>
MASTERADDRess
MASTERREAd
MASTERWRIte
MASTer
MATCHROM
MATH
MATHArbfltlt;x>
MATHFFTViewlt;n>
MATHFFTViewlt;x>
MATHFFTx
MATHlt;x>
MATHx
MATlab
MAX
MAXCAS
MAXCycle
MAXFRames
MAXHold
MAXIMUM
MAXPSRR
MAXPSRRFREQ
MAXREAD
MAXRETSIZE
MAXRTURN
MAXSamplerate
MAXTIMe
MAXVoltage
MAXWRITe
MAXimum
MDATA
MDATASpeed
ME
MEAN
MEANAUTOCalculate
MEANhistogram
MEAS
MEAS1
MEASRange
MEASTABle
MEASU
MEASUREAT
MEASUrement
MEASlt;x>
MEASured
MEASurement
MEASx
MEDian
MEDium
MEMORY
MESSaging
METHod
MID
MIDRef
MIL
MIN
MINCas
MINCycle
MINHold
MINIMUM
MINImum
MINMax
MINPSRR

MINPSRRFREQ
MINUI
MINimum
MISo
MIXED
MIXEDASCII
MIXEDHEX
MIXed
MKDir
MODE
MODEhistogram
MODe
MODel
MODulo
MOREEQua
MOREEQual
MOREEQualINrange
MORELimit
MOREThan
MOREthan
MOSFET
MOSi
MOUNT
MOVEABLE
MSB
MULTiply
MULtipleframes
Manual
Mathlt;x>
Mathx
N
NACK
NAK
NAME
NAMe
NAND
NANd
NATive
NAVigate
NDUTY
NDUTYCYCLE
NDUtY
NDUty
NEGAtive
NEGative
NETWORKCONFig
NEWpass
NEXT
NIBBLECount
NIBBLe
NO
NOCARE
NOISEAdd
NOISErej
NOISe
NOMINALOFFset
NOMinal
NOMinalfreq
NONE
NONTRANsition
NONe
NOPARity
NOR
NORESPonse
NORMALColor
NORMal
NORmal
NOSTATion
NOTE
NOTEQuals
NOVERSHOOT
NOne
NPERIOD
NPJ
NPULSE
NPULSEWIDTH
NR1
NR2
NR3
NRZ
NR_Pt
NTIMES
NTIMes
NULL
NULLFRDynamic
NULLFRStatic
NULl
NUMACQs
NUMACq
NUMAVg
NUMBer
NUMBins
NUMCHANnel
NUMERICORDer
NUMFRAMESACQuired
NUMOFOutputs
NUMOUTputs
NUMSEQuence
NWIDTH
NYET
None
OC1
OC12
OC3
OC48
OCCURS
ODD
ODDEVen
OFF
OFF1
OFFSet
ON
ONCE
ONE
ONEOVERDELTATVALUE
ONEPAIRI
ONEPAIRV
ONEPAIRVI
ONEWIRe
ONFAIL
OPCODEERRor
OPCode
OPERating
OPPositeas
OPTIMIZation
OPTIMize
OPTIONALPARam
OPTion
OPTional
OR
OUT
OUTEDGEQUALifier
OUTFILTers
OUTL2LTOL2N
OUTLINESelected
OUTPUT
OUTPUT1
OUTPUT1SOURce
OUTPUT1Type
OUTPUT1VOLTage
OUTPUT2
OUTPUT2SOURce
OUTPUT2Type
OUTPUT2VOLTage
OUTPUT3
OUTPUT3SOURce
OUTPUT3Type
OUTPUT3VOLTage
OUTPUT4
OUTPUT4SOURce
OUTPUT4VOLTage
OUTPUT5
OUTPUT5SOURce
OUTPUT5VOLTage
OUTPUT6
OUTPUT6SOURce
OUTPUT6VOLTage
OUTPUT7
OUTPUT7SOURce
OUTPUT7VOLTage
OUTPUTSOurce
OUTPUTType
OUTPUTlt;n>VOLTage
OUTPWR
OUTPWRSUM
OUTPut
OUTSIDEGreater
OUTSIDErange
OUTSOURCE
OUTWIRing
OUTrange
OUTside
OVERDRIVEMATCh
OVERDRIVESKIP
OVERDRIVe
OVERLoad
OVERRide
OVERlay
OVErdrive
OVErlay
Output1Pwr
Output2Pwr
Output3Pwr
P1W2V1I1
P1W3V2I2
P3W3
P3W3V2I2
P3W3V3I3
P3W4
PACKET
PACKets
PALEtte
PANKNOB
PARItyerror
PARallel
PARity
PASSFAILENabled
PASSFAILHIGHlimit
PASSFAILLIMit
PASSFAILLOWlimit
PASSFAILMARgin
PASSFAILWHEN
PASSWord
PATTERNDETECTION
PATTERNLENgth
PATTERNTYPe
PATTern
PAUSE
PAUSEPULSe
PAUSe
PAYLENgth
PAYLOAD

PAYLength
PAYLoad
PAYload
PCIE_GEN1
PCIE_GEN2
PCIE_GEN3
PCIExpress
PDUTTY
PDUTY
PDUTYCYCLE
PEAK
PEAKCURRent
PEAKVOLTage
PEAKdetect
PERCent
PERFREQ
PERIOD
PERIod
PERSistence
PERSource
PFACtor
PFC
PHASE
PHASECROSSOVERFREQ
PHASENOISE
PHASEONE
PHASETHREE
PHASETWO
PHASOR
PHASe
PHYSICALADDRess
PHYSicaladdress
PICPARameter
PID
PING
PIXELNUMBer
PIXELVALue
PIXel
PJ
PK2PK
PK2Pk
PKPK
PLL
PLOT
PLOTData
PLOTVIEWlt;x>
PLOTVIEWx
PLOTViewlt;n>
PLOTViewlt;x>
PLOTlt;x>
PLOTx
PM
PNG
PNJUNCTION
POHCL
POHCM
POHCS
POINT
POINTS
POLARITY
POLarity
POPUlation
PORT
PORTConfiguration
PORTReset
POSITIVe
POST
POSTAMBLE
POSition
POSitive
POVERSHOOT
POWERFACtor
POWERQUALITY
POWERRating
POWERx
POWLINear
POWLOG
POWer
POWerlt;x>
PPD
PPS101010
PPS121212
PPS565
PPS666
PPS888
PPULSE
PPULSEWIDTH
PREAMBLE
PRESET
PRESS
PREViewstate
PREVious
PREamble
PREsence
PRIMARYTURNs
PRIOD
PROBECOntrol
PROBECal
PROBEFunc
PROFile
PROTOCol
PROTOcol
PROTocol
PRObe
PSIFIVe
PSRR
PTYPe
PT_Fmt
PT_Off
PULSEWIDTh
PULSEWidth
PULSe
PULse
PWIDTH
PWRFACTOR
PWRFREQ
PWRUpstatus
ParaltBlockgt
Permeability
QFACTOR
QString
QStringNR1
QStringQString
QTAG
QTAGGING
QUALifier
QUAlifier
Qstring
RADians
RAIL17
RAILNUM
RAMP
RAMPtime
RANDom
RAP
RAPDATa
RATE100K
RATE10K
RATE10M
RATE115K
RATE11M
RATE125K
RATE12M
RATE13M
RATE14M
RATE153K
RATE15M
RATE16M
RATE19K
RATE1K
RATE1M
RATE20K
RATE250K
RATE25K
RATE2K
RATE2M
RATE300
RATE31K
RATE33K
RATE38K
RATE3M
RATE400K
RATE4K
RATE4M
RATE500K
RATE50K
RATE5M
RATE62K
RATE68K
RATE6M
RATE7M
RATE800K
RATE83K
RATE8M
RATE921K
RATE92K
RATE9K
RATE9M
RATEd
RAW10
RAW12
RAW14
RAW16
RAW20
RAW6
RAW7
RAW8
RBW
RBWMode
RCURRent
RDS
RDSON
RDSOn
READ
READFile
READROM
READout
READy
REAL
RECAll
RECAllmask
RECORD
RECORDLength
RECOrdlength
RECTANGLE
RECTANGular
RECTangle
RECTangular
RED
REF
REFERence
REFFFTViewlt;n>

REFFFTViewlt;x>
REFFFTx
REFLEVELMode
REFLevel
REFLevels
REFLevelslt;x>
REFMode
REFOUT
REFVOLTAGElt;n>Val
REFVoltage
REFerence
REFlt;x>
REFlt;x>_DALL
REFlt;x>_Dlt;x>
REFx
REFx_Dx
REGOWRIte
REGAddrtestconfg
REGAddrvendor
REGDatatestconfg
REGDatavendor
REGISTERADDRess
REGREAd
REGWRIte
REGion
REGister
REGisteraddress
REJ
REJect
REM
REMote
REName
REPEATERHOSt
REPEATERPERIPHERAL
REPEATStart
REPEATstart
REPOrt
REPWR
REPWRSUM
REPeating
REQDISConnect
REQSETINIT
REQUested
RERUN
RESAMPLE
RESAMPLErate
RESERVed
RESET
RESETTRIGger
RESOlution
RESPONSECODe
RESPonse
RESPonsetime
RESUME
RESUlt
RESUlts
RESample
RESet
RESistance
RF_AVErage
RF_FREQuency
RF_FREQuencylt;x>
RF_MAGnitude
RF_MAGnitudelt;x>
RF_MAXHold
RF_MINHold
RF_NORMal
RF_PHASe
RF_PHASelt;x>
RFvsTime
RGB444
RGB555
RGB565
RGB666
RGB888
RI
RIBinary
RIGht
RIO125
RIO250
RIO3125
RIPPLEFREQlt;n>Val
RISE
RISEHigh
RISELow
RISEMid
RISESLEWRATE
RISETIME
RISING
RISe
RISing
RJ
RJDD
RJDIRAC
RMDir
RMS
RNR
ROLL
ROLOCATION
ROMCODe
ROSc
ROTATion
RP
RPBinary
RR
RRBDisplaystate
RS232C
RSTDya
RUN
RUNSTop
RUNT
RUNt
RWINADDR
RX
S8B10B
SADDress
SAME
SAMEas
SAMPLERate
SAMPLEpoint
SAMple
SAS12_NOSSC
SAS12_SSC
SAS15_NOSSC
SAS15_SSC
SAS3_NOSSC
SAS3_SSC
SAS6_NOSSC
SAS6_SSC
SATA_GEN1
SATA_GEN2
SATA_GEN3
SAVEIMAGe
SAVEON
SAVEONEVent
SAVEWAVEform
SAVe
SAVemask
SC
SCALE
SCALERATio
SCALe
SCAle
SCIentific
SCLk
SCOPEApp REBOOT
SCRAMBLING
SCREEN
SCRambling
SDATa
SDI
SDIDATA
SDIDATASSM
SDLC
SDR
SDRBroadcast
SDRDirect
SE
SEARCH
SEARCH1
SEARCHOPTion
SEARCHROM
SEARCHSource
SEARCHTABle
SEARCHTable
SEARCHlt;x>
SEARCHtotrigger
SEARCHx>
SEARch
SEC1SOURce
SEC1TURNs
SEC2SOURce
SEC2TURNs
SEC3SOURce
SEC3TURNs
SEC4SOURce
SEC4TURNs
SEC5SOURce
SEC5TURNs
SEC6SOURce
SEC6TURNs
SECOND
SECOnds
SECPhase
SECVolt
SECWINDings
SEConds
SEGlt;x>
SEGlt;x>COUNT
SEGlt;y>
SELECTED
SELECTIONtype
SELTrace
SELect
SELected
SENSORADDRess
SENSORECU
SENSORSTATus
SENSor
SENT
SEQSETup
SEQnum
SEQuence
SERIALNUMBer
SERVICE
SERVICEMODe
SERVICENAMe
SERnumber
SESsion
SET
SETBrt
SETDecay

SETDya
SETFast
SETHOLDLogicval
SETHold
SETLevel
SETMrdl
SETMwrl
SETNdya
SETPs
SETRegaddr
SETRegdata
SETSlow
SETTime
SETUP
SETUp
SETWp
SEText
SEVEN
SFD
SFPbinary
SHAPE
SHORt
SHOWAReas
SHOWCRiteria
SHOWEQuation
SHOrt
SHUTDOWN
SHUTdown
SI
SIGNAL
SIGNALFREQUEncy
SIGNALTYpe
SIGNALType
SIGNal
SIGnal
SINC
SINE
SINGLE
SINX
SIX
SIXBIT
SIXFIFTyhz
SIXTEENeight
SIXTYHZ
SIXTyhz
SIXty
SIZE
SIZe
SKEW
SKIPROM
SLAVE
SLAVEADDRess
SLAVe
SLEEP
SLEWRATEMethod
SLEep
SLOW
SLOWERthan
SLOWer
SLOpe
SMPS
SNAp
SNRM
SNRME
SOA
SOAHITSCNT
SOCKETServer
SOF
SOFFRAMENUMber
SOT
SOURCE
SOURCEEDGEType
SOURCELIst
SOURCETYpe
SOURCEs
SOURce
SOUrce
SOUrceaddr
SOUrcelt;x>
SOUrceport
SOW
SPACEWIRe
SPAN
SPANABovebw
SPANBELowdc
SPANRBWRatio
SPEC
SPECIALPacket
SPECIALType
SPECTRAL
SPECTRALBUJ
SPECTRUM
SPECTral
SPECViewlt;x>
SPECtral
SPECtrum
SPI
SPLIT
SPLITMODE
SPLit
SPREADSheet
SPace
SQUELCH
SQUare
SRIbinary
SRJ
SRPbinary
SRQ
SS
SSC
SSCFREQDEV
SSCMODRATE
SSCPROFILE
SSM
SSPLIT
STADDress
STALL
STANDard
STANdard
START
STARTBIT
STARTCONDition
STARTFREQUEncy
STARTFREQuency
STARTFrequency
STARTPACKet
STARToffframe
STARTtime
STARTup
STARTupnosync
STARt
STATE
STATEBYTe
STATIC
STATIONADDr
STATIstics
STATUS
STATe
STATic
STATus
STAYSHigh
STAYSLow
STAcked
STAndard
STDDev
STEPRESPONSE
STOP
STOPACQ
STOPAcq
STOPAfter
STOPFREQuency
STOPFrequency
STROBE
STROBESource
STRObe
STRobe
STYPe
STYle
STandard
STop
SUBADdress
SUBGROUP
SUBNETMask
SUBSF
SUBTYPe
SUBtract
SUMFrame
SUPPORTBYTe
SUPPress
SUPervisory
SUSPEND
SV
SWITCHING
SWITCHINGLOSS
SWITCHINGRIPPLE
SWL
SWLCONFIGType
SWRIPPKPK
SWRIPRMS
SYMB
SYMBol
SYMbol
SYMbols
SYMmetry
SYNC
SYNCBITS
SYNCFrame
SYNCMODe
SYNCSEARch
SYNCTHRESHold
SYNCfield
SYSTEM
Start
Stop
String
Symbol
TARGETBER
TBIT
TBITRDNeg
TBITRDPos
TBITVALue
TCKAVG
TCPHeader
TDM
TDMVALue
TEKEXPonential
TEKSecure
TEMPerature
TENBASET
TENNinety
TERMinal
TERmination

TEST
TESTCONNection
TESTMODe
TESTmode
TESt
TEXT
TF
THDF
THDR
THIRD
THREE
THREEPHASEAUTOset
THREESIXTyhz
THRESHold
THReshold
TICKTIME
TICKTOLerance
TICKs
TIE
TIEHISTOGRAM
TIESPECTRUM
TIETIMETREND
TIMECode
TIMEOUTSIDELEVEL
TIMEOut
TIMEStamp
TIMETOMAX
TIMETOMIN
TIMETREND
TIMINGMode
TIMe
TJBER
TNTRATIO
TO
TOEDGESEARCHDIRect
TOEdge
TOFFENRG
TOFFLOSS
TOKENPacket
TOKENType
TOLERance
TOLerance
TOLevel
TONENRG
TONLOSS
TOOTHGAP
TOP
TOSYmbol
TOTAL
TOTALEFFiciency
TOTaluptime
TOUCHSCReen
TRACK
TRAILER
TRANSferbusownership
TRANSition
TRANsition
TRANstion
TRAPezoid
TRAiler
TRBit
TRIAngle
TRIGLevel
TRIGger
TRIGgertosearch<!
TRPWR
TRPWRSUM
TRUEPWR
TRUe
TRack
TTL
TTLENRG
TTLLOSS
TURN
TURNOFFtime
TURNON
TURNONTIME
TURNONtime
TWELVEtwelve
TWENTYBIT
TWENtyeighty
TWENtyfive
TWO
TWOBIT
TWOCLOCKCYCLES
TWOFifty
TWOThousand
TX
TXRX
TXRXTHRESHold
TYPE
TYPE1
TYPE2
TYPe
TotalEfficiency
True
U
UACK
UI
ULPS
UNDERLine
UNDERline
UNDO
UNEQual
UNINSTALL
UNIQue
UNITINTERVAL
UNITINtervals
UNITIntervals
UNITs
UNIts
UNLock
UNLocked
UNMOUNT
UNNumbered
UNWRap
UP
UPDatenow
UPPERFREQuency
UPPer
UPPerthreshold
USAGe
USB3
USBDevice
USBTmc
USECLockedge
USEClockedge
USEGLOBAL
USEREIGHTB
UTCDELTa
V
V1X
V2X
VALUe
VALidate
VALue
VARpersist
VBArs
VCESat
VCFACTOR
VDIFFXOVR
VECtors
VERBose
VERTICES
VERTPOS
VERTSCALE
VERTical
VERsion
VGLevel
VIEW
VIEWStyle
VIEWTYpe
VISIBility
VISual
VLEVELAbs
VLEVELPct
VOLTAGE
VOLTage
VOUT1SOUrce
VOUT2SOUrce
VOUT3SOUrce
VRMS
VSOURce
VSOUrce
VSYNCEND
VSYNCSTART
VTERm
VUNIT
WAITTime
WAKEup
WAKeup
WAVEFORM
WAVEVIEW1
WAVEView
WAVEViewlt;x>
WAVEViewlt;y>
WAVEform
WAVEforms
WAVFrm
WEIGht
WFId
WFMOutpre
WFMTYPe
WHEn
WIDTH
WIDTHBER
WIDth
WINDOW
WINDOWLENgth
WINSCALe
WINdow
WIRing
WIThin
WORD
WORDCOUNt
WORDSIZe
WORDSel
WORDSize
WRAP
WRITE
WRITEFile
WRITe
Word
X
XAUI
XAUI_GEN2
XAXIS
XAXISUnits
XFF
XID
XINcr
XN

XPOS
XPOSition
XUNIT
XUNit
XY
XYZ
XYZY
XZEro
XZYZ
Y
YAXIS
YCBCR
YCBCR12
YCBCR16
YCBCR24
YES
YMUlt
YN
YOFf
YPOS
YPOSition
YUNIT
YUNit
YUV
YUV420B10
YUV420L8
YUV422B10
YUV8BIT
YXZX
YZEro
ZERo
ZN
ZONe
ZOOM
ZOOMNR1
ZOOMOVERride
ZOOMlt;x>
Zoom
_FREQ_VS_TIME
_MAG_VS_TIME
_PHASE_VS_TIME
_SV_AVErage
_SV_MAXHold
_SV_MINHold
_SV_NORMal
block_data
boolean
directory_path
file_path
file_pathdata
forward
harmonics
ltasccurvegtPara
name
new_working_directory_path
old_file_pathnew_file_path
packetOffData
parity
reverse
source_file_pathdestination_
sourcefileMASKx
sourcefiledestination
v1
v2
wfmwfm
x
x201cMASKxx201d
x201cNR3NR3NR3NR3NR3NR3N
x201cPOWERxx201d

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:HIGHLEVEL	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	9.9999999999916E-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:SMTPLOGIN	"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	"XXXXXXXXXXXXXXXXXXXXXXXX"
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	"XXXXXXXXXXXXXXXXXXXXXXXX"
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	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:BRSBIT	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	"XXXXXXXXXXX"
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	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:CONDITION	SYNCFIELD
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE	"XXXXXXXX"
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	"XXXXXXXX"
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	"XXXXXXXXXXXXXXXX"
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	"XXXXXXXXXXXXXXXXXXXXXXXX"
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	"XXXXXXXXXXXXXXXXXXXXXXXX"
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	"XXXXXXXXXXX"
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	"XXXXXXX"
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	"XXXXXXXXXXXXXXXX"
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:BRSBIT	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	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET	0
TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:WORD	EITHER
TRIGGER:A:BUS:B1:CAN:CONDITION	SOF
TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION	NOCARE
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	"XXXXXXXXXXX"
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	"XXXXXXXXXXXXXXXX"
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	"XXXXXXXXXXX"
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	NOCARE
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	"XXXXXXXXXXXXXXXX"
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.*

WFMOutpre? 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 WFMOutpre: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 WFMOutpre: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 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). To change this value to LSB, use the WFMOutpre: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. WFMOutpre: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 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 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 WFMOutpre: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 WFMOutpre: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 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. This is a query only. (If you would like to determine only this value, use the WFMOutpre: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 WFMOUTpre:YUNit 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 WFMOUTpre:YUNit 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
:DATa:SOUrce CH1_D5	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 1	
:HEADer 1	
:VERBose 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 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
:CURVe?	Returns the following values. Each value represents a data point: :CURVe 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0

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 WFMOUTpre:YUNit 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 WFMOUTpre:YUNit 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
:DATa:SOUrce CH1_D1	
:ACQuire:MAGnivu 0	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 4	
:HEADer 1	
:VERBose 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 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
:CURVe?	Returns the following values. Each value represents a data point: :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

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 WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 4	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 32	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 "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. 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 WFMOutpre: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: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 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
: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 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
:CURVe?	Returns the following values. Each value represents a data point: :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

NOTE. *The returned hexadecimal data values are formatted without leading zeroes.*

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 8	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 64	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.

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. 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 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: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 WFMOUTpre:YUNit 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	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 4	
:HEADer 1	
:VERBose 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 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,FB6E6,FB666,FB6E6,FB666,FB666, FB6E6,FB666,FB6E6,FB6E6,FB666,FB6E6,FB666

NOTE. *The returned hexadecimal data values are formatted without leading zeroes.*

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 4	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 32	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nrcommand. 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.)

WFMOutpre? 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 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 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	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 8	
:HEADer 1	
:VERBose 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 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

WFMOutpre? 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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre:YUNit? query.)
YMULT 1.5625E-3	This value indicates the multiplying factor to convert the data point values 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.)

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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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 WFMOutpre: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.
`:WFMOutpre:ENCdg SRIbinary`	Request data LSB first.
`:WFMOutpre:YMUlt?`	Apply 2*YMUlt scaling to I and Q samples returned by CURVe?

Command	Comment
:WFMOutpre?	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: :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
:CURVe?	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.

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 autoset 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?, 2-75
ACQuire:FASTAcq:PALEtte, 2-75
ACQuire:FASTAcq:STATE, 2-76
ACQuire:MAXSamplerate?, 2-76
ACQuire:MODe, 2-76
ACQuire:NUMACq?, 2-78
ACQuire:NUMAVg, 2-78
ACQuire:SEQuence:CURrent?, 2-78
ACQuire:SEQuence:MODe, 2-79
ACQuire:SEQuence:NUMSEQuence, 2-79
ACQuire:STATE, 2-80
ACQuire:STOPAfter, 2-81
ACTONEVent:ENable, 2-81
ACTONEVent:LIMit, 2-82
ACTONEVent:LIMITCount, 2-82
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE, 2-83
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE, 2-83
ACTONEVent:MASKFail:ACTION:SRQ:STATE, 2-84
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE, 2-85
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE, 2-85
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE, 2-86
ACTONEVent:MASKHit:ACTION:SRQ:STATE, 2-87
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE, 2-87
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE, 2-88
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE, 2-89
ACTONEVent:MASKPass:ACTION:SRQ:STATE, 2-89
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE, 2-90
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE, 2-91
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE, 2-91
ACTONEVent:MEASUrement:ACTION:SRQ:STATE, 2-92
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE, 2-93
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE, 2-94
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE, 2-94
ACTONEVent:SEARCH:ACTION:SRQ:STATE, 2-95
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE, 2-96
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE, 2-96
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE, 2-97
ACTONEVent:TRIGger:ACTION:SRQ:STATE, 2-98
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE, 2-98
AFG:AMPLitude, 2-99
AFG:ARBitrary:SOUrce, 2-99
AFG:BURSt:CCOUnt, 2-100
AFG:BURSt:TRIGger, 2-100
AFG:FREQuency, 2-101
AFG:FUNCtion, 2-101
AFG:HIGHLevel, 2-102
AFG:LOWLevel, 2-102
AFG:NOISEAdd:PERCent, 2-103
AFG:NOISEAdd:STATE, 2-103
AFG:OFFSet, 2-104
AFG:OUTPut:LOAd:IMPEDance, 2-104
AFG:OUTPut:MODe, 2-105
AFG:OUTPut:STATE, 2-105
AFG:PERIod, 2-106
AFG:PULse:WIDth, 2-106
AFG:RAMP:SYMmetry, 2-107
AFG:SQUare:DUty, 2-107
ALIas, 2-108
ALIas:CATalog?, 2-108
ALIas:DEFine, 2-109
ALIas:DELEte, 2-110
ALIas:DELEte:ALL, 2-110
ALIas:DELEte:NAMe, 2-111
ALIas:STATE, 2-111

ALLEv?, 2-111
AUTOSAVEPITIMEOUT, 2-112
AUTOSAVEUITIMEOUT, 2-112
AUTOSet, 2-113
AUTOSet:ACQuisition:ENAble, 2-113
AUTOSet:ENAble, 2-114
AUTOSet:HORizontal:ENAble, 2-114
AUTOSet:TRIGger:ENAble, 2-115
AUTOSet:VERTical:ENAble, 2-116
AUTOSet:VERTical:OPTIMize, 2-116
AUXout:EDGE, 2-117
AUXout:SOUrce, 2-117

B

BATTery:ACPOWer?, 2-118
BATTery:SLOT<1,2>:CHARGE?, 2-118
BATTery:SLOT<1,2>:INSTalled?, 2-119
BATTery:SLOT<1,2>:SERIALnumber?, 2-119
BATTery:SLOT<1,2>:TIMETOEMPty?, 2-120
BATTery:SLOT<1,2>:TIMETOFULL?, 2-120
BUS:ADDNew, 2-121
BUS:B<x>:CAN:BITRate, 2-121
BUS:B<x>:CAN:BITRate:VALue, 2-122
BUS:B<x>:CAN:FD:BITRate, 2-122
BUS:B<x>:CAN:FD:BITRate:CUSTom, 2-123
BUS:B<x>:CAN:SAMPLEpoint, 2-124
BUS:B<x>:CAN:SIGNal, 2-124
BUS:B<x>:CAN:SOUrce, 2-125
BUS:B<x>:CAN:STANDard, 2-125
BUS:B<x>:CAN:THReshold, 2-126
BUS:B<x>:DISplay:FORMat, 2-126
BUS:B<x>:DISplay:LAYout, 2-127
BUS:B<x>:I2C:CLOCk:SOUrce, 2-128
BUS:B<x>:I2C:CLOCk:THReshold, 2-129
BUS:B<x>:I2C:DATa:SOUrce, 2-129
BUS:B<x>:I2C:DATa:THReshold, 2-130
BUS:B<x>:I2C:RWINADDR, 2-130
BUS:B<x>:LABel:COLor, 2-131
BUS:B<x>:LABel:FONT:BOLD, 2-132
BUS:B<x>:LABel:FONT:ITALic, 2-132
BUS:B<x>:LABel:FONT:SIZE, 2-133
BUS:B<x>:LABel:FONT:TYPE, 2-133
BUS:B<x>:LABel:FONT:UNDERline, 2-134
BUS:B<x>:LABel:name, 2-134
BUS:B<x>:LABel:XPOS, 2-135
BUS:B<x>:LABel:YPOS, 2-135
BUS:B<x>:LIN:BITRate, 2-136
BUS:B<x>:LIN:BITRate:CUSTom, 2-136
BUS:B<x>:LIN:IDFORmat, 2-137
BUS:B<x>:LIN:POLarity, 2-138
BUS:B<x>:LIN:SAMPLEpoint, 2-138
BUS:B<x>:LIN:SOUrce, 2-139
BUS:B<x>:LIN:SOUrce:THReshold, 2-139
BUS:B<x>:LIN:STANDard, 2-140
BUS:B<x>:PARallel:ALLTHResholds, 2-141
BUS:B<x>:PARallel:ALLTHResholds:APPly, 2-141
BUS:B<x>:PARallel:BIT<x>SOUrce, 2-142
BUS:B<x>:PARallel:BIT<x>SOUrce:
THReshold, 2-142
BUS:B<x>:PARallel:CLOCk:EDGE, 2-143
BUS:B<x>:PARallel:CLOCk:ISCLOCKED, 2-144
BUS:B<x>:PARallel:CLOCkSOUrce, 2-144
BUS:B<x>:PARallel:CLOCkSOUrce:
THReshold, 2-145
BUS:B<x>:RS232C:BITRate, 2-146
BUS:B<x>:RS232C:BITRate:CUSTom, 2-146
BUS:B<x>:RS232C:DATABits, 2-147
BUS:B<x>:RS232C:DELIMiter, 2-147
BUS:B<x>:RS232C:DISplaymode, 2-148
BUS:B<x>:RS232C:PARity, 2-149
BUS:B<x>:RS232C:POLarity, 2-149
BUS:B<x>:RS232C:SOUrce, 2-150
BUS:B<x>:RS232C:SOUrce:THReshold, 2-151
BUS:B<x>:SENT:CHANWidth, 2-151
BUS:B<x>:SENT:NIBBLECount, 2-152
BUS:B<x>:SENT:NUMCHANnel, 2-152
BUS:B<x>:SENT:PAUSEPULSe, 2-153
BUS:B<x>:SENT:POLARITY, 2-153
BUS:B<x>:SENT:SLOW, 2-154
BUS:B<x>:SENT:SOUrce, 2-155
BUS:B<x>:SENT:THRESHold, 2-155
BUS:B<x>:SENT:TICKTIME, 2-156
BUS:B<x>:SENT:TICKTOLerance, 2-156
BUS:B<x>:SPI:BITOrder, 2-157
BUS:B<x>:SPI:CLOCk:POLarity, 2-158
BUS:B<x>:SPI:CLOCk:SOUrce, 2-158
BUS:B<x>:SPI:CLOCk:THReshold, 2-159
BUS:B<x>:SPI:DATa:POLarity, 2-160
BUS:B<x>:SPI:DATa:SIZe, 2-160
BUS:B<x>:SPI:DATa:SOUrce, 2-161
BUS:B<x>:SPI:DATa:THReshold, 2-162
BUS:B<x>:SPI:FRAMING, 2-162
BUS:B<x>:SPI:IDLETime, 2-163
BUS:B<x>:SPI:MISo:DATa:POLarity, 2-163

BUS:B<x>:SPI:MISo:INPut, 2-164
BUS:B<x>:SPI:MISo:THReshold, 2-165
BUS:B<x>:SPI:MOSi:DATa:POLarity, 2-165
BUS:B<x>:SPI:MOSi:INPut, 2-166
BUS:B<x>:SPI:MOSi:THReshold, 2-166
BUS:B<x>:SPI:NUMBer:INputs, 2-167
BUS:B<x>:SPI:SELect:POLarity, 2-168
BUS:B<x>:SPI:SELect:SOUrce, 2-168
BUS:B<x>:SPI:SELect:THReshold, 2-169
BUS:B<x>:TYPe, 2-170
BUS:DELete, 2-170
BUS:LIST?, 2-171
BUSTABle:ADDNew, 2-171
BUSTABle:DELete, 2-171
BUSTABle:LIST?, 2-172
BUSY?, 2-172

C

*CAL?, 2-173
CALibrate?, 2-173
CALibrate:FACtory:STATus?, 2-173
CALibrate:INTERNal, 2-174
CALibrate:INTERNal:DATE?, 2-174
CALibrate:INTERNal:STARt, 2-175
CALibrate:INTERNal:STATus?, 2-175
CALibrate:PWRUpstatus?, 2-176
CALLOUTS:ADDNew, 2-176
CALLOUTS:CALLOUT<x>:BOOKMark:
SOURCE, 2-177
CALLOUTS:CALLOUT<x>:BOOKMark:
XPOS, 2-177
CALLOUTS:CALLOUT<x>:COLOR, 2-178
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:
X, 2-178
CALLOUTS:CALLOUT<x>:DISPLAYPOSition:
Y, 2-178
CALLOUTS:CALLOUT<x>:FONT:BOLD, 2-179
CALLOUTS:CALLOUT<x>:FONT:ITALIC, 2-179
CALLOUTS:CALLOUT<x>:FONT:SIZE, 2-180
CALLOUTS:CALLOUT<x>:FONT:TYPE, 2-180
CALLOUTS:CALLOUT<x>:FONT:
UNDERLine, 2-181
CALLOUTS:CALLOUT<x>:TEXT, 2-181
CALLOUTS:CALLOUT<x>:TYPE, 2-181
CALLOUTS:DELete, 2-182
CH<x>?, 2-182
CH<x>:BANdwidth, 2-183
CH<x>:CLIPping?, 2-183
CH<x>:COUPling, 2-184
CH<x>:DESKew, 2-185
CH<x>:DITHERrange, 2-185
CH<x>:INVert, 2-186
CH<x>:LABel:COLor, 2-186
CH<x>:LABel:FONT:BOLD, 2-187
CH<x>:LABel:FONT:ITALic, 2-187
CH<x>:LABel:FONT:SIZE, 2-188
CH<x>:LABel:FONT:TYPE, 2-188
CH<x>:LABel:FONT:UNDERline, 2-189
CH<x>:LABel:NAMe, 2-189
CH<x>:LABel:XPOS, 2-190
CH<x>:LABel:YPOS, 2-190
CH<x>:OFFSet, 2-191
CH<x>:POSition, 2-191
CH<x>:PROBEFunc:EXTAtten, 2-192
CH<x>:PROBEFunc:EXTDBatten, 2-192
CH<x>:PROBEFunc:EXTUnits, 2-193
CH<x>:PROBEFunc:EXTUnits:STATE, 2-193
CH<x>:SCAle, 2-194
CH<x>:SCALERATio, 2-194
CH<x>:TERmination, 2-195
CH<x>:VTERm:BIAS, 2-195
CLEAR, 2-196
*CLS, 2-196
Command Groups, 2-11
CONFIGuration:ANALOg:BANDWidth?, 2-197
CONNected:REQUested:STATus , 2-197
CONNected:STATus?, 2-198
CONNected:USAGe:TRack:REQUested:
STATus, 2-198
CONNected:USAGe:TRack:STATus?, 2-199
CURVe?, 2-200
CURVEStream?, 2-202
CUSTOMTABle:ADDNew, 2-199
CUSTOMTABle:DELete, 2-200
CUSTOMTABle:LIST?, 2-200

D

DATa, 2-204
DATa:ENCdg, 2-205
DATa:MODe, 2-207
DATa:RESample, 2-207
DATa:SOUrce, 2-208
DATa:SOUrce:AVAILable?, 2-208
DATa:STARt, 2-209

DATa:STOP, 2-210
DATa:WIDth, 2-211
DATE?, 2-211
DCH<x>_D<x>:LABel:COLor, 2-212
DCH<x>_D<x>:LABel:FONT:BOLD, 2-213
DCH<x>_D<x>:LABel:FONT:ITALic, 2-213
DCH<x>_D<x>:LABel:FONT:SIZE, 2-214
DCH<x>_D<x>:LABel:FONT:TYPE, 2-214
DCH<x>_D<x>:LABel:FONT:UNDERline, 2-215
DCH<x>_D<x>:LABel:NAMe, 2-215
DCH<x>_D<x>:THReshold, 2-212
*DDT, 2-216
DESE, 2-217
DIAg:LOOP:OPTion, 2-218
DIAg:LOOP:OPTion:NTIMes, 2-218
DIAg:LOOP:STOP, 2-218
DIAg:MODe, 2-219
DIAg:RESUlt?, 2-219
DIAg:RESUlt:FLAg?, 2-220
DIAg:RESUlt:LOG?, 2-220
DIAg:SELect, 2-220
DIAg:STATE, 2-221
DISplay?, 2-221
DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor, 2-222
DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor, 2-223
DISplay:COLors, 2-224
DISplay:GLOBal:B<x>:STATE, 2-224
DISplay:GLOBal:CH<x>:STATE, 2-225
DISplay:GLOBal:DCH<x>:STATE, 2-225
DISplay:GLOBal:MATH<x>:STATE, 2-226
DISplay:GLOBal:REF<x>:STATE, 2-226
DISplay:INTENSITy?, 2-227
DISplay:INTENSITy:BACKLight, 2-227
DISplay:INTENSITy:BACKLight:AUTODim:ENAble, 2-228
DISplay:INTENSITy:BACKLight:AUTODim:TIMe, 2-228
DISplay:MATHFFTView<x>:AUTOScale, 2-230
DISplay:MATHFFTView<x>:CURSor:ASOUrce?, 2-230
DISplay:MATHFFTView<x>:CURSor:BSOUrce?, 2-231
DISplay:MATHFFTView<x>:CURSor:DDT?, 2-231
DISplay:MATHFFTView<x>:CURSor:FUNCtion, 2-231
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition, 2-232
DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts?, 2-233
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition, 2-233
DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts?, 2-233
DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa?, 2-234
DISplay:MATHFFTView<x>:CURSor:MODe, 2-234
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE?, 2-235
DISplay:MATHFFTView<x>:CURSor:ROLOCATION, 2-229
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition, 2-235
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition, 2-236
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition, 2-236
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition, 2-237
DISplay:MATHFFTView<x>:CURSor:STATE, 2-237
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition, 2-238
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts?, 2-238
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition, 2-239
DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts?, 2-239
DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa?, 2-240
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition, 2-240
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition, 2-241
DISplay:MATHFFTView<x>:GRIDlines, 2-241
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE, 2-242
DISplay:MATHFFTView<x>:XAXIS:SCALE, 2-242
DISplay:MATHFFTView<x>:YAXIS:SCALE, 2-243
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM, 2-243
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO, 2-244

DISplay:MATHFFTView<x>:ZOOM:YAXIS:
FROM, 2-244
DISplay:MATHFFTView<x>:ZOOM:YAXIS:
TO, 2-245
DISplay:PERSistence, 2-245
DISplay:PERSistence:RESET, 2-246
DISplay:PLOTView<x>:AUTOScale, 2-247
DISplay:PLOTView<x>:CURSor:ASOUrce?, 2-248
DISplay:PLOTView<x>:CURSor:BSOUrce?, 2-248
DISplay:PLOTView<x>:CURSor:DDT?, 2-248
DISplay:PLOTView<x>:CURSor:FUNCtion, 2-249
DISplay:PLOTView<x>:CURSor:HBArs:
APOSition, 2-249
DISplay:PLOTView<x>:CURSor:HBArs:
AUNIts?, 2-250
DISplay:PLOTView<x>:CURSor:HBArs:
BPOSition, 2-250
DISplay:PLOTView<x>:CURSor:HBArs:
BUNIts?, 2-251
DISplay:PLOTView<x>:CURSor:HBArs:
DELTa?, 2-251
DISplay:PLOTView<x>:CURSor:MODe, 2-252
DISplay:PLOTView<x>:CURSor:
ONEOVERDELTATVALUE?, 2-252
DISplay:PLOTView<x>:CURSor:
ROLOCATION, 2-246
DISplay:PLOTView<x>:CURSor:SCREEN:
AXPOSition, 2-253
DISplay:PLOTView<x>:CURSor:SCREEN:
AYPOSition, 2-253
DISplay:PLOTView<x>:CURSor:SCREEN:
BXPOSition, 2-254
DISplay:PLOTView<x>:CURSor:SCREEN:
BYPOSition, 2-254
DISplay:PLOTView<x>:CURSor:
SPLITMODE, 2-255
DISplay:PLOTView<x>:CURSor:STATE, 2-255
DISplay:PLOTView<x>:CURSor:VBArs:
APOSition, 2-256
DISplay:PLOTView<x>:CURSor:VBArs:
BPOSition, 2-256
DISplay:PLOTView<x>:CURSor:VBArs:
DELTa?, 2-257
DISplay:PLOTView<x>:CURSor:VBArs:
UNIts?, 2-257
DISplay:PLOTView<x>:CURSor:WAVEform:
APOSition, 2-258
DISplay:PLOTView<x>:CURSor:WAVEform:
BPOSition, 2-258
DISplay:PLOTView<x>:GRIDlines, 2-259
DISplay:PLOTView<x>:XAXIS:SCALE, 2-260
DISplay:PLOTView<x>:YAXIS:SCALE, 2-260
DISplay:PLOTView<x>:ZOOM:XAXIS:
FROM, 2-261
DISplay:PLOTView<x>:ZOOM:XAXIS:TO, 2-261
DISplay:PLOTView<x>:ZOOM:YAXIS:
FROM, 2-262
DISplay:PLOTView<x>:ZOOM:YAXIS:TO, 2-262
DISplay:REFFFTView<x>:AUTOScale, 2-263
DISplay:REFFFTView<x>:CURSor:
ASOUrce?, 2-264
DISplay:REFFFTView<x>:CURSor:
BSOUrce?, 2-264
DISplay:REFFFTView<x>:CURSor:DDT?, 2-265
DISplay:REFFFTView<x>:CURSor:
FUNCtion, 2-265
DISplay:REFFFTView<x>:CURSor:HBArs:
APOSition, 2-266
DISplay:REFFFTView<x>:CURSor:HBArs:
AUNIts?, 2-266
DISplay:REFFFTView<x>:CURSor:HBArs:
BPOSition, 2-267
DISplay:REFFFTView<x>:CURSor:HBArs:
BUNIts?, 2-267
DISplay:REFFFTView<x>:CURSor:HBArs:
DELTa?, 2-267
DISplay:REFFFTView<x>:CURSor:MODe, 2-268
DISplay:REFFFTView<x>:CURSor:
ONEOVERDELTATVALUE?, 2-268
DISplay:REFFFTView<x>:CURSor:
ROLOCATION, 2-263
DISplay:REFFFTView<x>:CURSor:SCREEN:
AXPOSition, 2-269
DISplay:REFFFTView<x>:CURSor:SCREEN:
AYPOSition, 2-269
DISplay:REFFFTView<x>:CURSor:SCREEN:
BXPOSition, 2-270
DISplay:REFFFTView<x>:CURSor:SCREEN:
BYPOSition, 2-270
DISplay:REFFFTView<x>:CURSor:
SPLITMODE, 2-271
DISplay:REFFFTView<x>:CURSor:STATE, 2-271
DISplay:REFFFTView<x>:CURSor:VBArs:
APOSition, 2-272

DISplay:REFFFTView<x>:CURSor:VBArs:
BPOSition, 2-272
DISplay:REFFFTView<x>:CURSor:VBArs:
DELTa?, 2-273
DISplay:REFFFTView<x>:CURSor:VBArs:
UNIts?, 2-273
DISplay:REFFFTView<x>:CURSor:WAVEform:
AHPOSition?, 2-274
DISplay:REFFFTView<x>:CURSor:WAVEform:
APOSition, 2-274
DISplay:REFFFTView<x>:CURSor:WAVEform:
AVPOSition?, 2-274
DISplay:REFFFTView<x>:CURSor:WAVEform:
BHPOSition?, 2-275
DISplay:REFFFTView<x>:CURSor:WAVEform:
BPOSition, 2-275
DISplay:REFFFTView<x>:CURSor:WAVEform:
BVPOSition?, 2-276
DISplay:REFFFTView<x>:GRIDlines, 2-276
DISplay:REFFFTView<x>:REF:REF<x>:
STATE, 2-277
DISplay:REFFFTView<x>:XAXIS:SCALE, 2-277
DISplay:REFFFTView<x>:ZOOM:XAXIS:
FROM, 2-278
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO, 2-278
DISplay:REFFFTView<x>:ZOOM:YAXIS:
FROM, 2-279
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO, 2-279
DISplay:SELect:BUS, 2-280
DISplay:SELect:MATH, 2-280
DISplay:SELect:REFerence, 2-281
DISplay:SELect:SOUrce, 2-281
DISplay:SELect:VIEW, 2-282
DISplay:SELect:WAVEView<x>:SOUrce, 2-282
DISplay:VARpersist, 2-283
DISplay:WAVEform, 2-283
DISplay:WAVEView:CURSor:CURSOR1:
ROLOCATION, 2-284
DISplay:WAVEView:GRIDTYPE, 2-284
DISplay:WAVEView<x>:BUS:B<x>:STATE, 2-285
DISplay:WAVEView<x>:BUS:B<x>:VERTical:
POSition, 2-285
DISplay:WAVEView<x>:CH<x>:STATE, 2-286
DISplay:WAVEView<x>:CH<x>:VERTical:
POSition, 2-287
DISplay:WAVEView<x>:CH<x>:VERTical:
SCAle, 2-287
DISplay:WAVEView<x>:CURSor?, 2-288
DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:AVPOSition?, 2-302
DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:BVPOSition?, 2-303
DISplay:WAVEView<x>:CURSor:
CURSOR<x>?, 2-288
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ASOUrce, 2-289
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
BSOUrce, 2-290
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
DDT?, 2-290
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
FUNCtion, 2-291
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:APOSition, 2-292
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:AUNIts?, 2-292
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:BPOSition, 2-293
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:BUNIts?, 2-293
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:DELTa?, 2-294
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
MODe, 2-294
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ONEOVERDELTATVALUE?, 2-295
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AXPOSition, 2-295
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AYPOSition, 2-296
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BXPOSition, 2-297
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BYPOSition, 2-297
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SPLITMODE, 2-298
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
STATE, 2-298
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:APOSition, 2-299
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:BPOSition, 2-300
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:DELTa?, 2-300

DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:UNIts?, 2-301
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:APOSition, 2-301
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:BPOSition, 2-302
DISplay:WAVEView<x>:DCH<x>_D<x>:
STATE, 2-303
DISplay:WAVEView<x>:DCH<x>_DALL:
DIGORDER, 2-304
DISplay:WAVEView<x>:DCH<x>_DALL:VERTical:
POSition, 2-305
DISplay:WAVEView<x>:FILTer, 2-305
DISplay:WAVEView<x>:GRAticule, 2-306
DISplay:WAVEView<x>:INTENSITy:
GRATicule, 2-306
DISplay:WAVEView<x>:INTENSITy:
WAVEform, 2-307
DISplay:WAVEView<x>:MATH:MATH<x>:
AUTOScale, 2-307
DISplay:WAVEView<x>:MATH:MATH<x>:
STATE, 2-308
DISplay:WAVEView<x>:MATH:MATH<x>:
VERTical:POSition, 2-308
DISplay:WAVEView<x>:MATH:MATH<x>:
VERTical:SCAle, 2-309
DISplay:WAVEView<x>:REF:REF<x>:
STATE, 2-309
DISplay:WAVEView<x>:REF:REF<x>:VERTical:
POSition, 2-310
DISplay:WAVEView<x>:REF:REF<x>:VERTical:
SCAle, 2-311
DISplay:WAVEView<x>:STYle, 2-311
DISplay:WAVEView<x>:VIEWStyle, 2-312
DISplay:WAVEView<x>:ZOOM?, 2-312
DISplay:WAVEView<x>:ZOOM:ZOOM<x>?, 2-313
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:POSition, 2-313
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:SCALe, 2-314
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:WINSCALe, 2-315
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
STATe, 2-315
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
VERTical:POSition, 2-316
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
VERTical:SCALe, 2-317
DISplay:WAVEView<y>:REF:REF<x>:
FRAMe, 2-317
DISplay:WAVEView<y>:REF<x>_DALL:
FRAMe, 2-318

E

*ESE, 2-318
*ESR?, 2-319
ETHERnet:DHCPbootp, 2-320
ETHERnet:DNS:IPADDress, 2-320
ETHERnet:DOMAINname, 2-321
ETHERnet:ENET:ADDress?, 2-321
ETHERnet:GATEWay:IPADDress, 2-321
ETHERnet:IPADDress, 2-322
ETHERnet:LXI:LAN:RESET, 2-322
ETHERnet:LXI:LAN:SERVICENAMe, 2-323
ETHERnet:LXI:LAN:STATus?, 2-323
ETHERnet:NAME, 2-324
ETHERnet:NETWORKCONFig, 2-324
ETHERnet:PING, 2-325
ETHERnet:PING:STATus?, 2-325
ETHERnet:SUBNETMask, 2-325
EVENT?, 2-326
EVMsg?, 2-327
EVQty?, 2-327

F

FACtory, 2-328
FILESystem?, 2-329
FILESystem:COPy, 2-329
FILESystem:CWD, 2-330
FILESystem:DELEte, 2-331
FILESystem:DIR?, 2-331
FILESystem:HOMEDir?, 2-332
FILESystem:LDIR?, 2-332
FILESystem:MKDir, 2-333
FILESystem:MOUNT:DRIVE, 2-333
FILESystem:MOUNT:TEKDrive, 2-334
FILESystem:READFile, 2-335
FILESystem:REName, 2-336
FILESystem:RMDir, 2-336
FILESystem:TEKDrive:CODE?, 2-337
FILESystem:TEKDrive:CODE:EXPirytime?, 2-337
FILESystem:TEKDrive:CODE:STATus?, 2-337

FILESystem:UNMOUNT:DRIve, 2-338
FILESystem:UNMOUNT:TEKDrive, 2-338
FILESystem:WRITEFile, 2-339
FPAnel:PRESS, 2-339
FPAnel:TURN, 2-340

H

HEADer, 2-340
HORizontal?, 2-341
HORizontal:ACQDURATION?, 2-342
HORizontal:DELay:MODe, 2-342
HORizontal:DELay:TIMe, 2-343
HORizontal:DIVisions?, 2-343
HORizontal:MAIn:INTERPRatio?, 2-343
HORizontal:MODE, 2-344
HORizontal:MODE:MANual:CONFIGure, 2-344
HORizontal:MODE:RECOrdlength, 2-345
HORizontal:MODE:SAMPLERate, 2-345
HORizontal:MODE:SCAle, 2-346
HORizontal:POSition, 2-346
HORizontal:PREViewstate?, 2-347
HORizontal:RECOrdlength, 2-347
HORizontal:SAMPLERate, 2-348
HORizontal:SAMPLERate:ANALYZemode:
MINimum:OVERRide, 2-348
HORizontal:SAMPLERate:ANALYZemode:
MINimum:VALue, 2-349
HORizontal:SCAle, 2-349

I

ID?, 2-350
*IDN?, 2-350

L

LIC:UNINSTALL?, 2-350
LICense?, 2-351
LICense:APPID?, 2-351
LICense:COUNt?, 2-352
LICense:ERRor?, 2-352
LICense:GMT?, 2-353
LICense:HID?, 2-353
LICense:INSTall, 2-353
LICense:ITEM?, 2-354
LICense:LIST?, 2-354
LICense:VALidate?, 2-355
LOCk, 2-355
*LRN?, 2-356

M

MAINWindow:BADGe:BRINgtoview, 2-357
MAINWindow:FONTSize, 2-357
MAINWindow:RRBDisplaystate, 2-358
MASK:DELete, 2-358
MASK:MASK<x>:COUNT?, 2-358
MASK:MASK<x>:COUNT:HITS?, 2-359
MASK:MASK<x>:COUNT:SEG<x>:HITS?, 2-361
MASK:MASK<x>:DEFinedby, 2-359
MASK:MASK<x>:DISplay, 2-360
MASK:MASK<x>:LIST?, 2-360
MASK:MASK<x>:SEG<x>:POINTS, 2-361
MASK:MASK<x>:SOUrce, 2-362
MASK:MASK<x>:TESt:CTHReshold, 2-362
MASK:MASK<x>:TESt:STATE, 2-363
MASK:MASK<x>:TESt:STATUS?, 2-363
MASK:MASK<x>:TESt:THReshold, 2-364
MASK:MASK<x>:TOLerance:HABSolute, 2-364
MASK:MASK<x>:TOLerance:HORizontal, 2-365
MASK:MASK<x>:TOLerance:UPDatenow, 2-366
MASK:MASK<x>:TOLerance:VABSolute, 2-366
MASK:MASK<x>:TOLerance:VERTical, 2-367
MASK:MASK<x>:TTYPe, 2-367
MASK:TESt:WAVEforms, 2-368
MATH:ADDNew, 2-369
MATH:DELete, 2-369
MATH:LIST?, 2-370
MATH:MATH<x>:AVG:MODE, 2-370
MATH:MATH<x>:AVG:WEIGht, 2-371
MATH:MATH<x>:CAN:SUPPortedfields, 2-371
MATH:MATH<x>:DEFine, 2-372
MATH:MATH<x>:FUNCtion, 2-372
MATH:MATH<x>:GATing, 2-374
MATH:MATH<x>:I2C:SUPPortedfields, 2-373
MATH:MATH<x>:INTERpolation, 2-373
MATH:MATH<x>:LABel:COLor, 2-374
MATH:MATH<x>:LABel:FONT:BOLD, 2-375
MATH:MATH<x>:LABel:FONT:ITALic, 2-375
MATH:MATH<x>:LABel:FONT:SIZE, 2-376
MATH:MATH<x>:LABel:FONT:TYPE, 2-376
MATH:MATH<x>:LABel:FONT:UNDERline, 2-377
MATH:MATH<x>:LABel:NAMe, 2-377
MATH:MATH<x>:LABel:XPOS, 2-378
MATH:MATH<x>:LABel:YPOS, 2-378
MATH:MATH<x>:LIN:SUPPortedfields, 2-379

MATH:MATH<x>:PARallel:SUPPortedfields, 2-379
MATH:MATH<x>:RS232C:SUPPortedfields, 2-380
MATH:MATH<x>:SENT:SUPPortedfields, 2-380
MATH:MATH<x>:SIGNeddata, 2-381
MATH:MATH<x>:SOUrce<x>, 2-382
MATH:MATH<x>:SPECTral:HORZ, 2-382
MATH:MATH<x>:SPECTral:MAG, 2-383
MATH:MATH<x>:SPECTral:PHASE, 2-383
MATH:MATH<x>:SPECTral:SOUrce, 2-384
MATH:MATH<x>:SPECTral:SUPPress, 2-384
MATH:MATH<x>:SPECTral:SUPPress:VALue, 2-385
MATH:MATH<x>:SPECTral:TYPE, 2-386
MATH:MATH<x>:SPECTral:UNWRap, 2-386
MATH:MATH<x>:SPECTral:UNWRap:DEGrees, 2-387
MATH:MATH<x>:SPECTral:WINdow, 2-387
MATH:MATH<x>:SPI:SUPPortedfields, 2-388
MATH:MATH<x>:TYPe, 2-389
MATH:MATH<x>:VUNIT, 2-390
MEASTABle:ADDNew, 2-390
MEASTABle:DELETE, 2-390
MEASTABle:LIST?, 2-391
MEASUrement?, 2-391
MEASUrement:ADDMEAS , 2-392
MEASUrement:ADDNew, 2-395
MEASUrement:ANNOTate, 2-395
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh, 2-396
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow, 2-396
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid, 2-397
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis, 2-397
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh, 2-398
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow, 2-398
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid, 2-399
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE, 2-399
MEASUrement:CH<x>:REFLevels:BASETop, 2-400
MEASUrement:CH<x>:REFLevels:METHod, 2-400
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh, 2-401
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow, 2-401
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid, 2-402
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis, 2-402
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh, 2-403
MEASUrement:CH<x>:REFLevels:PERCent:RISELow, 2-403
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid, 2-404
MEASUrement:CH<x>:REFLevels:PERCent:TYPE, 2-405
MEASUrement:DELete , 2-405
MEASUrement:DELETEALL , 2-405
MEASUrement:EDGE<x>, 2-406
MEASUrement:GATing, 2-406
MEASUrement:GATing:ACTive, 2-407
MEASUrement:GATing:ENDtime, 2-407
MEASUrement:GATing:HYSTeresis, 2-408
MEASUrement:GATing:LOGICSource, 2-408
MEASUrement:GATing:MIDRef, 2-409
MEASUrement:GATing:SEARCHSource, 2-409
MEASUrement:GATing:STARTtime, 2-410
MEASUrement:INTERp, 2-410
MEASUrement:LIST?, 2-411
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh, 2-411
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow, 2-412
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid, 2-412
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis, 2-413
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh, 2-413
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow, 2-414
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid, 2-414
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE, 2-415
MEASUrement:MATH<x>:REFLevels:BASETop, 2-415
MEASUrement:MATH<x>:REFLevels:METHod, 2-416

MEASUrement:MATH<x>:REFLevels:PERCent:
FALLHigh, 2-417
MEASUrement:MATH<x>:REFLevels:PERCent:
FALLLow, 2-417
MEASUrement:MATH<x>:REFLevels:PERCent:
FALLMid, 2-418
MEASUrement:MATH<x>:REFLevels:PERCent:
HYSTeresis, 2-418
MEASUrement:MATH<x>:REFLevels:PERCent:
RISEHigh, 2-419
MEASUrement:MATH<x>:REFLevels:PERCent:
RISELow, 2-419
MEASUrement:MATH<x>:REFLevels:PERCent:
RISEMid, 2-420
MEASUrement:MATH<x>:REFLevels:PERCent:
TYPE, 2-420
MEASUrement:MEAS<x>:BURSTEDGTYPe, 2-421
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MAXimum?, 2-421
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MEAN?, 2-422
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MINimum?, 2-422
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
PK2PK?, 2-423
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
POPUlation?, 2-423
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
STDDev?, 2-424
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MAXimum?, 2-424
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MEAN?, 2-424
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MINimum?, 2-425
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:PK2PK?, 2-425
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:POPUlation?, 2-426
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:STDDev?, 2-426
MEASUrement:MEAS<x>:CYCLemode, 2-426
MEASUrement:MEAS<x>:DELay:EDGE<x>, 2-427
MEASUrement:MEAS<x>:DISPlaystat:
ENABle, 2-428
MEASUrement:MEAS<x>:EDGE<x>, 2-431
MEASUrement:MEAS<x>:EDGEIncre, 2-428
MEASUrement:MEAS<x>:EDGES:
FROMLevel, 2-429
MEASUrement:MEAS<x>:EDGES:LEVel, 2-429
MEASUrement:MEAS<x>:EDGES:N, 2-430
MEASUrement:MEAS<x>:EDGES:
SLEWRATEMethod, 2-430
MEASUrement:MEAS<x>:EDGES:TOLevel, 2-431
MEASUrement:MEAS<x>:FAILCount?, 2-432
MEASUrement:MEAS<x>:FROMedge, 2-433
MEASUrement:MEAS<x>:
FROMEDGESEARCHDIRect, 2-433
MEASUrement:MEAS<x>:GATing, 2-434
MEASUrement:MEAS<x>:GATing:ACTive, 2-435
MEASUrement:MEAS<x>:GATing:ENDtime, 2-435
MEASUrement:MEAS<x>:GATing:GLOBal, 2-436
MEASUrement:MEAS<x>:GATing:
HYSTeresis, 2-436
MEASUrement:MEAS<x>:GATing:
LOGICSource, 2-437
MEASUrement:MEAS<x>:GATing:MIDRef, 2-437
MEASUrement:MEAS<x>:GATing:
SEARCHSource, 2-438
MEASUrement:MEAS<x>:GATing:
STARTtime, 2-438
MEASUrement:MEAS<x>:GLOBalref, 2-439
MEASUrement:MEAS<x>:HIGHREFVoltage, 2-439
MEASUrement:MEAS<x>:IDLETime, 2-440
MEASUrement:MEAS<x>:LABel, 2-440
MEASUrement:MEAS<x>:LOWREFVoltage, 2-441
MEASUrement:MEAS<x>:
PASSFAILENabled, 2-441
MEASUrement:MEAS<x>:
PASSFAILHIGHlimit, 2-442
MEASUrement:MEAS<x>:PASSFAILLIMit, 2-443
MEASUrement:MEAS<x>:
PASSFAILLOWlimit, 2-444
MEASUrement:MEAS<x>:
PASSFAILMARgin, 2-444
MEASUrement:MEAS<x>:PASSFAILWHEN, 2-445
MEASUrement:MEAS<x>:PERFREQ:EDGE, 2-446
MEASUrement:MEAS<x>:POLarity, 2-447
MEASUrement:MEAS<x>:REFLevels:ABSolute:
FALLHigh, 2-447
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:FALLLow, 2-448
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:FALLMid, 2-448

MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:HYSTeresis, 2-449
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISEHigh, 2-449
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISELow, 2-450
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISEMid, 2-450
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:TYPE, 2-451
MEASUrement:MEAS<x>:REFLevels<x>:
BASETop, 2-452
MEASUrement:MEAS<x>:REFLevels<x>:
METHod, 2-452
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLHigh, 2-453
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLLow, 2-454
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLMid, 2-454
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
HYSTeresis, 2-455
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISEHigh, 2-455
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISELow, 2-456
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISEMid, 2-456
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
TYPE, 2-457
MEASUrement:MEAS<x>:REFMode, 2-458
MEASUrement:MEAS<x>:REFVoltage, 2-458
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MAXimum?, 2-459
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MEAN?, 2-459
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MINimum?, 2-459
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
PK2PK?, 2-460
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
POPUlation?, 2-460
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
STDDev?, 2-461
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MAXimum?, 2-461
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MEAN?, 2-462
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MINimum?, 2-462
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
PK2PK?, 2-463
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
POPUlation?, 2-463
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
STDDev?, 2-463
MEASUrement:MEAS<x>:SIGNALType, 2-464
MEASUrement:MEAS<x>:SOUrce<x>, 2-464
MEASUrement:MEAS<x>:STATUS?, 2-465
MEASUrement:MEAS<x>:TOEdge, 2-466
MEASUrement:MEAS<x>:
TOEDGESEARCHDIRect, 2-466
MEASUrement:MEAS<x>:TRANSition, 2-467
MEASUrement:MEAS<x>:TYPe, 2-468
MEASUrement:MEAS<x>:XUNIT?, 2-470
MEASUrement:MEAS<x>:YUNIT?, 2-471
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLHigh, 2-481
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLLow, 2-482
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLMid, 2-482
MEASUrement:REF<x>:REFLevels:ABSolute:
HYSTeresis, 2-483
MEASUrement:REF<x>:REFLevels:ABSolute:
RISEHigh, 2-483
MEASUrement:REF<x>:REFLevels:ABSolute:
RISELow, 2-484
MEASUrement:REF<x>:REFLevels:ABSolute:
RISEMid, 2-484
MEASUrement:REF<x>:REFLevels:ABSolute:
TYPE, 2-485
MEASUrement:REF<x>:REFLevels:
BASETop, 2-485
MEASUrement:REF<x>:REFLevels:METHod, 2-486
MEASUrement:REF<x>:REFLevels:PERCent:
FALLHigh, 2-486
MEASUrement:REF<x>:REFLevels:PERCent:
FALLLow, 2-487
MEASUrement:REF<x>:REFLevels:PERCent:
FALLMid, 2-487
MEASUrement:REF<x>:REFLevels:PERCent:
HYSTeresis, 2-488

MEASUrement:REF<x>:REFLevels:PERCent:
RISEHigh, 2-488
MEASUrement:REF<x>:REFLevels:PERCent:
RISELow, 2-489
MEASUrement:REF<x>:REFLevels:PERCent:
RISEMid, 2-489
MEASUrement:REF<x>:REFLevels:PERCent:
TYPE, 2-490
MEASUrement:REFLevels:ABSolute:
FALLHigh, 2-471
MEASUrement:REFLevels:ABSolute:
FALLLow, 2-472
MEASUrement:REFLevels:ABSolute:
FALLMid, 2-472
MEASUrement:REFLevels:ABSolute:
HYSTeresis, 2-473
MEASUrement:REFLevels:ABSolute:
RISEHigh, 2-473
MEASUrement:REFLevels:ABSolute:
RISELow, 2-474
MEASUrement:REFLevels:ABSolute:
RISEMid, 2-474
MEASUrement:REFLevels:ABSolute:TYPE, 2-474
MEASUrement:REFLevels:BASETop, 2-475
MEASUrement:REFLevels:METHod, 2-475
MEASUrement:REFLevels:MODE, 2-476
MEASUrement:REFLevels:PERCent:
FALLHigh, 2-477
MEASUrement:REFLevels:PERCent:
FALLLow, 2-477
MEASUrement:REFLevels:PERCent:
FALLMid, 2-478
MEASUrement:REFLevels:PERCent:
HYSTeresis, 2-478
MEASUrement:REFLevels:PERCent:
RISEHigh, 2-479
MEASUrement:REFLevels:PERCent:
RISELow, 2-479
MEASUrement:REFLevels:PERCent:
RISEMid, 2-480
MEASUrement:REFLevels:PERCent:TYPE, 2-480
MEASUrement:REFLevels:TYPE, 2-481
MEASUrement:STATIstics:CYCLEMode, 2-491

N

NEWpass, 2-491

O

*OPC, 2-492
*OPT?, 2-492

P

PASSWord, 2-493
PAUSe, 2-493
PG:AMPlitude, 2-494
PG:BIT:ONE, 2-494
PG:BIT:THREE, 2-495
PG:BIT:TWO, 2-496
PG:BIT:ZERO, 2-496
PG:BITRate, 2-497
PG:BURSt:CCOUnt, 2-497
PG:BURSt:TRIGger, 2-498
PG:FILE:PATTern, 2-498
PG:OUTPut:MODe, 2-499
PG:PATTERNdefinition, 2-499
PLOT:ADDNew, 2-500
PLOT:DELete, 2-500
PLOT:LIST?, 2-501
PLOT:PLOT<x>:SOUrce<x>, 2-501
PLOT:PLOT<x>:TYPe, 2-501
POWer:ADDNew, 2-502
POWer:DELete, 2-502
POWer:POWer<x>:CLRESPONSE:
AMP<x>Val, 2-503
POWer:POWer<x>:CLRESPONSE:
AMPMode, 2-503
POWer:POWer<x>:CLRESPONSE:
ANALYSISMethod?, 2-504
POWer:POWer<x>:CLRESPONSE:
CONNECTSTATus?, 2-505
POWer:POWer<x>:CLRESPONSE:
CONSTAMPlitude, 2-505
POWer:POWer<x>:CLRESPONSE:
FREQ<x>Val, 2-506
POWer:POWer<x>:CLRESPONSE:GENerator, 2-507
POWer:POWer<x>:CLRESPONSE:
GENIPADDress, 2-507
POWer:POWer<x>:CLRESPONSE:
IMPEDance, 2-508
POWer:POWer<x>:CLRESPONSE:
INPUTSOurce, 2-508
POWer:POWer<x>:CLRESPONSE:
OUTPUTSOurce, 2-509
POWer:POWer<x>:CLRESPONSE:PPD, 2-509

POWer:POWer<x>:CLRESPONSE:
STARTFREQuency, 2-510
POWer:POWer<x>:CLRESPONSE:
STOPFREQuency, 2-511
POWer:POWer<x>:CLRESPONSE:
TESTCONNection, 2-511
POWer:POWer<x>:PRESET, 2-512
POWer:POWer<x>:RESUlts:CURRentacq:
MAXimum?, 2-512
POWer:POWer<x>:RESUlts:CURRentacq:
MEAN?, 2-513
POWer:POWer<x>:RESUlts:CURRentacq:
MINimum?, 2-514
*PSC, 2-514
*PUD, 2-515

R

RECAll:MASK, 2-516
RECAll:SESsion, 2-516
RECAll:SETUp, 2-517
RECAll:WAVEform, 2-518
REF:ADDNew, 2-519
REF:DELete, 2-519
REF:LIST?, 2-519
REF:REF<x>:DESKew, 2-520
REF:REF<x>:LABel:COLor, 2-520
REF:REF<x>:LABel:FONT:BOLD, 2-521
REF:REF<x>:LABel:FONT:ITALic, 2-521
REF:REF<x>:LABel:FONT:SIZE, 2-522
REF:REF<x>:LABel:FONT:TYPE, 2-522
REF:REF<x>:LABel:FONT:UNDERline, 2-522
REF:REF<x>:LABel:NAMe, 2-523
REF:REF<x>:LABel:XPOS, 2-523
REF:REF<x>:LABel:YPOS, 2-524
REF:REF<x>:SOUrce, 2-524
REM, 2-525
ROSc:SOUrce, 2-525
ROSc:STATE?, 2-526
*RST, 2-526

S

SAVe:EVENTtable:BUS, 2-528
SAVe:EVENTtable:MEASUrement, 2-528
SAVe:EVENTtable:SEARCHTable, 2-529
SAVe:IMAGe, 2-529
SAVe:IMAGe:COMPosition, 2-530
SAVe:IMAGe:VIEWTYpe, 2-530
SAVe:MASK, 2-531
SAVe:PLOTData, 2-532
SAVe:REPOrt, 2-532
SAVe:REPOrt:COMMents, 2-533
SAVe:SESsion, 2-533
SAVe:SETUp, 2-534
SAVe:SETUp:INCLUDEREFs, 2-535
SAVe:WAVEform, 2-535
SAVe:WAVEform:GATing, 2-537
SAVe:WAVEform:GATing:RESAMPLErate, 2-538
SAVe:WAVEform:SOURCELIst?, 2-538
SAVEON:FILE:DEST, 2-539
SAVEON:FILE:NAME, 2-539
SAVEON:IMAGe, 2-540
SAVEON:IMAGe:FILEFormat, 2-541
SAVEON:TRIGger, 2-541
SAVEON:WAVEform, 2-542
SAVEON:WAVEform:FILEFormat, 2-543
SAVEON:WAVEform:SOURce, 2-544
SAVEONEVent:FILEDest, 2-544
SAVEONEVent:FILEName, 2-545
SAVEONEVent:IMAGe:FILEFormat, 2-545
SAVEONEVent:WAVEform:FILEFormat, 2-546
SAVEONEVent:WAVEform:SOUrce, 2-547
SCOPEApp REBOOT, 2-547
SEARCH:ADDNew, 2-548
SEARCH:DELete, 2-548
SEARCH:DELETEALL, 2-548
SEARCH:LIST?, 2-549
SEARCH:SEARCH<x>:COPy, 2-549
SEARCH:SEARCH<x>:NAVigate, 2-549
SEARCH:SEARCH<x>:TOTAL?, 2-550
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
CONDition, 2-550
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
DATa:DIRection, 2-551
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
DATa:OFFSet, 2-551
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
DATa:QUALifier, 2-552
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
DATa:SIZe, 2-553
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
DATa:VALue, 2-553
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:
ERRType, 2-554

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit, 2-555
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit, 2-555
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype, 2-556
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe, 2-557
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue, 2-557
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe, 2-558
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue, 2-558
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition, 2-559
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection, 2-560
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe, 2-560
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue, 2-561
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition, 2-562
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue, 2-562
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier, 2-563
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe, 2-563
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue, 2-564
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE, 2-564
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue, 2-565
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue, 2-566
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition, 2-566
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe, 2-567
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue, 2-567
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition, 2-568
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType, 2-569
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC, 2-569
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue, 2-570
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier, 2-571
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue, 2-572
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue, 2-572
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier, 2-573
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue, 2-574
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue, 2-575
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier, 2-576
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue, 2-577
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue, 2-577
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue, 2-578
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier, 2-579
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue, 2-580
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue, 2-581
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue, 2-581
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier, 2-582
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue, 2-583
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue, 2-584
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce, 2-584
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition, 2-585
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe, 2-585

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue, 2-586
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe, 2-586
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe, 2-587
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce, 2-588
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold, 2-588
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold, 2-589
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime, 2-589
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion, 2-590
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce, 2-590
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>, 2-591
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>, 2-591
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>, 2-592
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>, 2-592
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:DCH<x>_D<x>, 2-593
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>, 2-594
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>, 2-594
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity, 2-595
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge, 2-595
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn, 2-596
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit, 2-597
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification, 2-597
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit, 2-598
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity, 2-598
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce, 2-599
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold, 2-599
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn, 2-600
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification, 2-601
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity, 2-601
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce, 2-602
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH, 2-602
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW, 2-603
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn, 2-603
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth, 2-604
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE, 2-604
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce, 2-605
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold, 2-606
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime, 2-606
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>, 2-607
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>, 2-607
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>, 2-608
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>, 2-608
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:DCH<x>_D<x>, 2-609
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>, 2-609
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>, 2-610
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime, 2-611
SEARCH:SEARCH<x>:TRIGger:A:STATE, 2-611
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq, 2-612

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
LOGICQUALification, 2-612
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
POLarity, 2-613
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
SOUrce, 2-613
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
THReshold, 2-614
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:
TIMe, 2-614
SEARCH:SEARCH<x>:TRIGger:A:TYPe, 2-615
SEARCH:SELected, 2-616
SECurity:LOGger:STATe, 2-616
SELect:CH<x>, 2-617
SELect:DCH<x>:DAll, 2-618
SET?, 2-618
SOCKETServer:ENABle, 2-619
SOCKETServer:PORT, 2-620
SOCKETServer:PROTOCol, 2-620
*SRE, 2-621
*STB?, 2-621

T

TEKSecure, 2-622
TIMe, 2-623
TIMe:ZONe, 2-623
TIMe:ZONe:UTCDELTa, 2-624
TOTaluptime?, 2-624
TOUCHSCReen:STATe, 2-624
*TRG, 2-625
TRIGger, 2-625
TRIGger:A, 2-626
TRIGger:A:BUS:B<x>:CAN:CONDition, 2-626
TRIGger:A:BUS:B<x>:CAN:DATa:DIRection, 2-627
TRIGger:A:BUS:B<x>:CAN:DATa:OFFSet, 2-628
TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier, 2-629
TRIGger:A:BUS:B<x>:CAN:DATa:SIZe, 2-630
TRIGger:A:BUS:B<x>:CAN:DATa:VALue, 2-630
TRIGger:A:BUS:B<x>:CAN:ERRType, 2-631
TRIGger:A:BUS:B<x>:CAN:FD:BRSBit, 2-631
TRIGger:A:BUS:B<x>:CAN:FD:ESIBit, 2-632
TRIGger:A:BUS:B<x>:CAN:FRAMEtype, 2-633
TRIGger:A:BUS:B<x>:CAN:IDentifier:MODe, 2-633
TRIGger:A:BUS:B<x>:CAN:IDentifier:
VALue, 2-634
TRIGger:A:BUS:B<x>:I2C:ADDRess:MODe, 2-634
TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue, 2-635
TRIGger:A:BUS:B<x>:I2C:CONDition, 2-636
TRIGger:A:BUS:B<x>:I2C:DATa:DIRection, 2-636
TRIGger:A:BUS:B<x>:I2C:DATa:SIZe, 2-637
TRIGger:A:BUS:B<x>:I2C:DATa:VALue, 2-637
TRIGger:A:BUS:B<x>:LIN:CONDition, 2-638
TRIGger:A:BUS:B<x>:LIN:DATa:HIVALue, 2-639
TRIGger:A:BUS:B<x>:LIN:DATa:QUALifier, 2-639
TRIGger:A:BUS:B<x>:LIN:DATa:SIZe, 2-640
TRIGger:A:BUS:B<x>:LIN:DATa:VALue, 2-641
TRIGger:A:BUS:B<x>:LIN:ERRTYPE, 2-641
TRIGger:A:BUS:B<x>:LIN:IDentifier:VALue, 2-642
TRIGger:A:BUS:B<x>:PARallel:DATa:VALue, 2-642
TRIGger:A:BUS:B<x>:RS232C:CONDition, 2-643
TRIGger:A:BUS:B<x>:RS232C:DATa:SIZe, 2-644
TRIGger:A:BUS:B<x>:RS232C:DATa:VALue, 2-644
TRIGger:A:BUS:B<x>:SENT:CONDition, 2-645
TRIGger:A:BUS:B<x>:SENT:ERRType, 2-645
TRIGger:A:BUS:B<x>:SENT:ERRType:CRC, 2-646
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
HIVALue, 2-647
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
QUALifier, 2-647
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN1A:
VALue, 2-648
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
HIVALue, 2-649
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
QUALifier, 2-650
TRIGger:A:BUS:B<x>:SENT:FAST:CHAN2B:
VALue, 2-651
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTer:
HIVALue, 2-651
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTer:
QUALifier, 2-652
TRIGger:A:BUS:B<x>:SENT:FAST:COUNTer:
VALue, 2-653
TRIGger:A:BUS:B<x>:SENT:FAST:INVERTNIBble:
VALue, 2-654
TRIGger:A:BUS:B<x>:SENT:FAST:STATus:
VALue, 2-655
TRIGger:A:BUS:B<x>:SENT:PAUSE:
QUALifier, 2-655
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
HIVALue, 2-656
TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
QUALifier, 2-657

TRIGger:A:BUS:B<x>:SENT:SLOW:DATA:
VALue, 2-658
TRIGger:A:BUS:B<x>:SENT:SLOW:IDentifier:
VALue, 2-659
TRIGger:A:BUS:B<x>:SPI:CONDition, 2-659
TRIGger:A:BUS:B<x>:SPI:DATa:SIZe, 2-660
TRIGger:A:BUS:B<x>:SPI:DATa:VALue, 2-660
TRIGger:A:BUS:SOUrce, 2-661
TRIGger:A:EDGE:COUPling, 2-661
TRIGger:A:EDGE:SLOpe, 2-662
TRIGger:A:EDGE:SOUrce, 2-663
TRIGger:A:HOLDoff:TIMe, 2-684
TRIGger:A:LEVel:CH<x>, 2-663
TRIGger:A:LOGIc:DELTatime, 2-664
TRIGger:A:LOGIc:FUNCtion, 2-664
TRIGger:A:LOGIc:INPut:CLOCk:SOUrce, 2-665
TRIGger:A:LOGIc:POLarity, 2-666
TRIGger:A:LOGIc:USECLockedge, 2-666
TRIGger:A:LOGIc:WHEn, 2-667
TRIGger:A:LOGICPattern:CH<x>, 2-667
TRIGger:A:LOGICPattern:DCH<x>_D<x>, 2-668
TRIGger:A:LOWerthreshold:CH<x>, 2-669
TRIGger:A:MODe, 2-684
TRIGger:A:PULSEWidth:HIGHLimit, 2-669
TRIGger:A:PULSEWidth:LOWLimit, 2-670
TRIGger:A:PULSEWidth:POLarity, 2-670
TRIGger:A:PULSEWidth:SOUrce, 2-671
TRIGger:A:PULSEWidth:WHEn, 2-671
TRIGger:A:RUNT:POLarity, 2-672
TRIGger:A:RUNT:SOUrce, 2-673
TRIGger:A:RUNT:WHEn, 2-674
TRIGger:A:RUNT:WIDth, 2-674
TRIGger:A:SETHold:CLOCk:EDGE, 2-675
TRIGger:A:SETHold:CLOCk:SOUrce, 2-675
TRIGger:A:SETHold:HOLDTime, 2-676
TRIGger:A:SETHold:SETTime, 2-676
TRIGger:A:SETHOLDLOGICVAL:
DCH<x>_D<x>, 2-677
TRIGger:A:TIMEOut:POLarity, 2-678
TRIGger:A:TIMEOut:SOUrce, 2-678
TRIGger:A:TIMEOut:TIMe, 2-679
TRIGger:A:TRANsition:DELTatime, 2-679
TRIGger:A:TRANsition:
LOGICQUALification, 2-680
TRIGger:A:TRANsition:POLarity, 2-680
TRIGger:A:TRANsition:SOUrce, 2-681
TRIGger:A:TRANsition:WHEn, 2-681
TRIGger:A:TYPe, 2-682
TRIGger:A:UPPerthreshold:CH<x>, 2-683
TRIGger:AUXLevel, 2-685
TRIGger:HYSTeresis:USER:STATe, 2-685
TRIGger:HYSTeresis:USER:VALue, 2-686
TRIGger:STATE?, 2-686
*TST?, 2-687

U

UNDO, 2-687
UNLock, 2-688
USBDevice:CONFigure, 2-688

V

VERBose, 2-689
VERTical:DESKew:FROM:
CUSTOMPROPAgation, 2-690
VERTical:DESKew:FROM:SOUrce, 2-690
VERTical:DESKew:STATIC, 2-691
VERTical:DESKew:TO:
CUSTOMPROPAgation, 2-691
VERTical:DESKew:TO:SOUrce, 2-692
VXI:ENAble, 2-692
VXI:PORT:HIGH, 2-693
VXI:PORT:LOW, 2-693

W

*WAI, 2-694
WAVFrm?, 2-694
WFMOutpre?, 2-695
WFMOutpre:ASC_Fmt?, 2-695
WFMOutpre:BIT_Nr, 2-696
WFMOutpre:BN_Fmt, 2-697
WFMOutpre:BYT_Nr, 2-697
WFMOutpre:BYT_Or, 2-698
WFMOutpre:CENTERFREQuency?, 2-698
WFMOutpre:DOMain?, 2-699
WFMOutpre:ENCdg, 2-699
WFMOutpre:NR_Pt?, 2-700
WFMOutpre:PT_Fmt?, 2-701
WFMOutpre:PT_Off?, 2-701
WFMOutpre:PT_ORder?, 2-702
WFMOutpre:RESample, 2-703
WFMOutpre:SPAN?, 2-703
WFMOutpre:WFId?, 2-704
WFMOutpre:WFMTYPe?, 2-705

WFMOutpre:XINcr?, 2-706
WFMOutpre:XUNit?, 2-706
WFMOutpre:XZEro?, 2-706
WFMOutpre:YMUlt?, 2-707
WFMOutpre:YOFf?, 2-708
WFMOutpre:YUNit?, 2-708
WFMOutpre:YZEro?, 2-708