

# DPO7000, DPO70000/B, DSA70000/B, and MSO70000 Series Digital Oscilloscopes Programmer Manual



077-0010-09

**Tektronix**



**DPO7000, DPO70000/B, DSA70000/B, and MSO70000 Series  
Digital Oscilloscopes  
Programmer Manual**

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.

FastFrame, OpenChoice, iView, Pinpoint, RT-Eye, MyScope, TekLink, TekVPI, and MultiView Zoom are trademarks of Tektronix, Inc.

DPO7000, DPO70000/B, DSA70000/B, and MSO70000 Series Programmer Online Help, 076-0021-09

## **Contacting Tektronix**

Tektronix, Inc.  
14200 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.tektronix.com](http://www.tektronix.com) to find contacts in your area.

---

# Table of Contents

Preface .....	iii
Getting Started .....	1-1
Setting Up Remote Communications .....	1-1
Command Syntax .....	2-1
Backus-Naur Form Notation .....	2-1
Command and Query Structure .....	2-1
Clearing the Instrument .....	2-3
Command Entry .....	2-4
Constructed Mnemonics .....	2-6
Argument Types .....	2-7
Command Groups .....	2-11
Acquisition Command Group .....	2-11
Alias Command Group .....	2-12
Bus Command Group .....	2-13
Calibration Command Group .....	2-15
Cursor Command Group .....	2-15
Diagnostics Command Group .....	2-18
Digital Command Group .....	2-19
Display Control Command Group .....	2-20
E-mail Command Group .....	2-23
Error Detector Command Group .....	2-25
File System Command Group .....	2-34
Hard Copy Command Group .....	2-35
Histogram Command Group .....	2-36
Horizontal Command Group .....	2-37
Limit Test Command Group .....	2-40
Low Speed Serial Trigger Command Group .....	2-42
Mask Command Group .....	2-45
Math Command Group .....	2-51
Measurement Command Group .....	2-53
Miscellaneous Command Group .....	2-57
Save and Recall Command Group .....	2-59
Search and Mark Command Group .....	2-60
Status and Error Command Group .....	2-69
TekLink Command Group .....	2-70
Trigger Command Group .....	2-71
Vertical Command Group .....	2-79
Waveform Transfer Command Group .....	2-84
Zoom Command Group .....	2-90

Commands Listed in Alphabetical Order .....	2-93
Status and Events .....	3-1
Registers .....	3-1
*PSC Command .....	3-4
Queues .....	3-4
Event Handling Sequence .....	3-5
Synchronization Methods .....	3-6
Appendix A: Character Set .....	A-1
Appendix B: Reserved Words .....	B-1
Appendix C: Factory Default Setup Values .....	C-1
Default Setup .....	C-1
Appendix D: GPIB Interface Specifications .....	D-1
Interface Messages .....	D-1
GPIB Functions .....	D-2
Glossary	
Index	

---

# Preface

This programmer manual provides you with the information required to use GPIB commands for remotely controlling your instrument.

## Documentation



**Quick Start User Manual.** The user manual has information about installing and operating the instrument. It also provides concepts and theories about using the instrument that are not covered in the online help.

**Online Help.** This is an online help system that is integrated with the User Interface application that ships with this product. The online help provides in-depth operation and user interface help.

**Getting Started with OpenChoice™ Solutions Manual.** A book that explores some options for getting data from your instrument into any one of several available analysis tools.

**Specifications and Performance Verification.** Instrument specifications and a performance verification procedure. This is available as a printable PDF file on the Product Software CD-ROM.

**TekVISA Programmer Manual.** This manual is available as a printable PDF file on the Product Software CD-ROM. The manual describes TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

**Optional Applications Software for Tektronix Windows-Based Instruments (CD-ROM).** This CD-ROM contains trial versions of application-specific programs that you can install and run five times per application. To purchase an application, contact your local Tektronix representative.

**Other Included Documentation.** Installation booklets are included in the Product Software and Operating System Restore Software packages.

**Service Manual.** The service manual includes procedures to service the instrument to the module level. This manual is available as a printable PDF file on the Product Software CD-ROM.

---

The programmer guide 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 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.
- **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.
- **Status and Events.** This topic discusses the status and event reporting system for the GPIB 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 may return, including error messages.
- **Miscellaneous.** This topic contains miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and interface specifications that may be helpful when using commands to remotely control the instrument.



---

# Getting Started

This programmer guide provides you with the information required to use GPIB commands for remotely controlling your instrument. With this information, you can write computer programs that will perform functions such as setting the front panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs, such as spreadsheets.

In addition to the traditional GPIB electronic interface (referred to as the physical GPIB interface), your instrument is provided with a *TekVISA* GPIB-compatible interface (referred to as the virtual GPIB interface). This is a software Application Programming Interface (API) which enables you to communicate with the instrument in a variety of ways, including via the Internet. With the following two exceptions, these interfaces are completely independent:

- **HEADER.** Command headers enabled or disabled on one interface are correspondingly enabled or disabled on the other interface. Refer to the command descriptions for more detailed information.
- **VERBOSE.** Verbosity enabled or disabled on one interface is correspondingly enabled or disabled on the other interface. Refer to the command description for more detailed information.

Most examples in this document require that both **HEADER** and **VERBOSE** are **ON**.

Refer to *Documentation* for information on related manuals and documents.

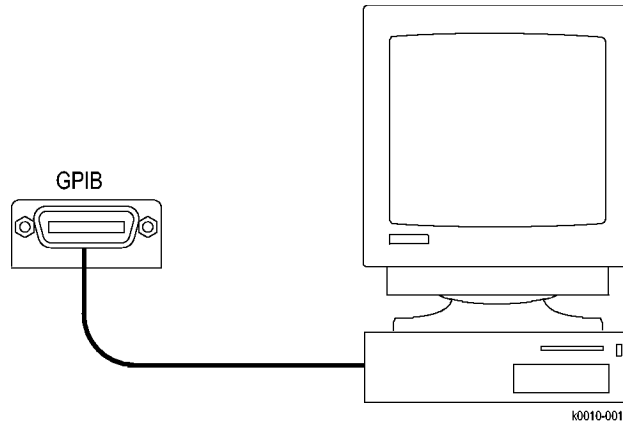
## Setting Up Remote Communications

Before setting up the instrument for remote communications using the electronic (physical) GPIB interface, you should familiarize yourself with the following GPIB requirements:

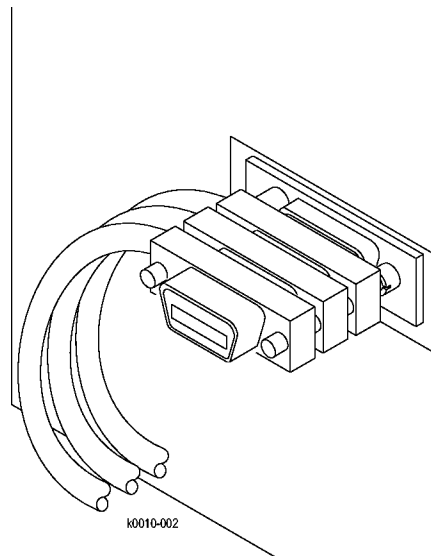
- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No more than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No more than 65 feet (20 meters) of cable should be used to connect devices to a bus.
- At least two-thirds of the devices on the network should be powered on while using the network.
- Connect the devices on the network in a star or linear configuration. Do not use loop or parallel configurations.

### Connecting to the Instrument

Your instrument has a 24-pin GPIB connector on its rear (side) panel. This connector has a D-type shell and conforms to IEEE Std 488.1<sup>3</sup>/<sub>4</sub>1987. Attach an IEEE Std 488.1<sup>3</sup>/<sub>4</sub>1987 GPIB cable to this connector and to your controller as shown in the following figure.



If necessary, the GPIB connectors can be stacked as shown in the following figure.



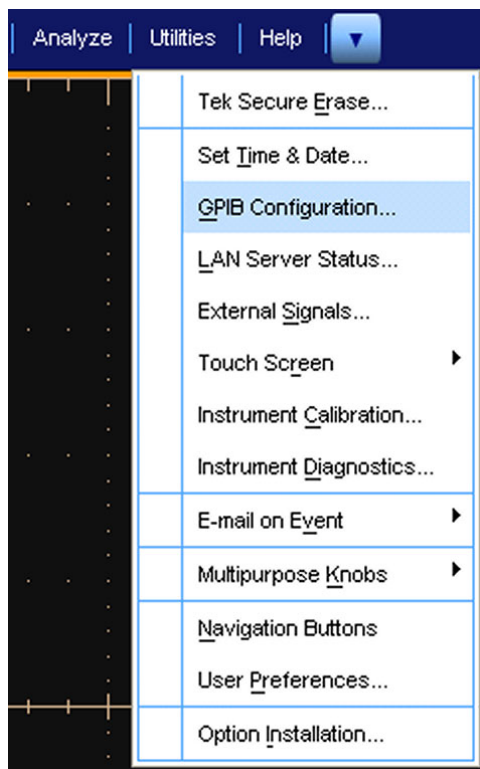
### Setting the GPIB Address

To function correctly, your instrument must have a unique device address. The default settings for the GPIB configuration are:

- GPIB Address 1
- GPIB Mode Talk/Listen

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

1. Select GPIB Configuration from the Utilities menu.



2. Click the Configuration Talk/Listen button.



3. Change the GPIB Address to a unique address.
4. Click the Close button.

The instrument is now set up for bidirectional communication with your controller.



# Command Syntax

You can control the operations and functions of the instrument through the GPIB interface using commands and queries. The following related topics listed 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
( )	Comment

## Command and Query Structure

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

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark on 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**

<b>Symbol</b>	<b>Meaning</b>
<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 an asterisk (*).
<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:

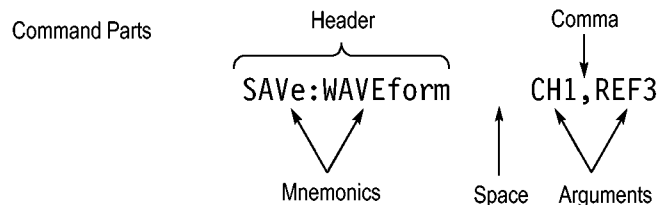
- [ : ] <Header> ?
- [ : ] <Header> ? [ <Space> <Argument> [ <Comma> <Argument> ] . . . ]

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. For example, HISTogram:STATistics:STDdev? returns the standard deviation of the histogram, while HISTogram:STATistics? returns all the histogram statistics, and HISTogram? returns all the histogram parameters.

**Headers** You can control whether the instrument returns headers as part of the query response. Use the HEADer command to control this feature. 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 following table shows the difference in responses.

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

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



## Clearing the Instrument

You can clear the Output Queue and reset the instrument to accept a new command or query by using the selected Device Clear (DCL) GPIB function. Refer to your GPIB library documentation for further details about the selected Device Clear operation.

## 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 abbreviations in capitals. For example, you can enter the command ACQure: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 10`, can be concatenated into the following single command:

```
TRIGger:MODE NORMAl;:ACQuire:NUMAVg 10
```

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 10` into a single command:

```
ACQuire:MODE ENVeloPe; NUMAVg 10
```

The longer version works equally well:

```
ACQuire:MODE ENVeloPe;:ACQuire:NUMAVg 10
```

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

```
ACQuire:MODE ENVeloPe;*OPC
```

Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODE ENVeloPe;*OPC;NUMAVg 10` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 10.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display imageview color is temperature and the display recordview color is spectral, the concatenated query `DISPlay:COLOR:PALETTE:IMAGEVIEW?;RECORDVIEW?` will return the following.

If the header is on:

```
:DISPLAY:COLOR:PALETTE:IMAGEVIEW TEMPERATURE;
:DISPLAY:COLOR:PALETTE:RECORDVIEW SPECTRAL
```

If the header is off:

```
TEMPERATURE;SPECTRAL
```

5. 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:PERsistance:RESET;ACQuire:NUMAVg 10 (no colon before ACQuire)

DISplay:GRaticule FULL;:FILTer SINX (extra colon before FILTer; use DISplay:GRaticule FULL;FILTer SINX instead)

DISplay:PERsistance:RESET;:\*OPC (colon before a star (\*) command)

DISplay:COLOr:MATHCOLOr DEFAULT;COLOr:REFCOLOr INHERIT (levels of the mnemonics are different; either remove the second use of COLOr or place :DISplay: in front of COLOr:REFCOLOr INHERIT)

**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 linefeed (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, or CH4. 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>.

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

**Table 2-5: Cursor Mnemonics**

Symbol	Meaning
CURSOR<x>	A cursor selector; <x> is either 1 or 2.
POSITION<x>	A cursor selector; <x> is either 1 or 2.
HPOS<x>	A cursor selector; <x> is either 1 or 2.

**Math Specifier Mnemonics** Commands can specify the mathematical waveform to use as a mnemonic in the header.

**Table 2-6: Math Specifier Mnemonics**

Symbol	Meaning
Math<x>	A math waveform specifier; <x> is 1 through 4.

**Measurement Specifier Mnemonics** Commands can specify which measurement to set or query as a mnemonic in the header. Up to eight automated measurements may be displayed.

**Table 2-7: Measurement Specifier Mnemonics**

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is 1 through 8.

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

**Table 2-8: Channel Mnemonics**

Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 through 4.

**Reference Waveform Mnemonics** Commands can specify the reference waveform to use as a mnemonic in the header.

**Table 2-9: Reference Waveform Mnemonics**

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is 1 thru 4.

## Argument Types

**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 follows:

**Table 2-10: Numeric Arguments**

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent

Most numeric arguments will be automatically forced to a valid setting, either by 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 2-11: 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 if 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. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.
6. A carriage return or line feed embedded in a quoted string does not terminate the string, but is treated as just another character in the string.
7. The maximum length of a quoted string returned from a query is 255 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**

Several instrument commands use a block argument form (see the following table).

**Table 2-12: Block Argument**

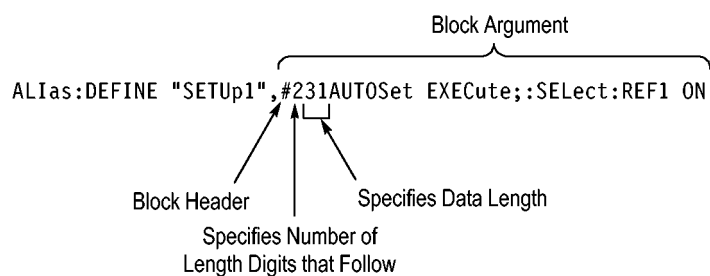
Symbol	Meaning
<NZDig>	A nonzero digit character in the range of 1–9
<Dig>	A digit character, in the range of 0–9

Table 2-12: Block Argument (cont.)

Symbol	Meaning
<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.

**NOTE.** The digit <NZDig> is in hexadecimal format. This deviates slightly from the IEEE 488.2 specification that it be in decimal format, as extra allowances must be made for data lengths that are greater than 999,999,999 <DChar> elements (for example 500M record lengths at 2 bytes per point).





# Command Groups

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

## Acquisition Command Group

Use the commands in the Acquisition Command Group to set up the modes and functions that control how the instrument acquires the signals you input to the channels 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.
- Determine the action the system takes upon completing an acquisition, such as saving all waveforms and taking a measurement when the acquisition is stopped.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

**Table 2-13: Acquisition Commands**

Command	Description
<a href="#">ACquire?</a>	Returns acquisition parameters
<a href="#">ACquire:INTERPEightbit</a>	Sets or queries the interpolation acquisition mode
<a href="#">ACquire:MODE</a>	Sets or queries acquisition mode
<a href="#">ACquire:NUMFRAMESACquired?</a>	Returns the number of acquisitions that have occurred
<a href="#">ACquire:NUMACq?</a>	Returns the number of waveform acquisitions that have occurred since starting acquisition with the ACquire:STATE RUN command
<a href="#">ACquire:NUMAVg</a>	Sets or queries number of acquisitions for an averaged waveform
<a href="#">ACquire:NUMEnv</a>	Sets or queries number of acquisitions for envelope waveform
<a href="#">ACquire:NUMSAMPles</a>	Sets or queries the number of samples that make up a WfmDB for single sequence mode and Mask Pass/Fail Completion Test

Table 2-13: Acquisition Commands (cont.)

Command	Description
<a href="#">ACQUIRE:SAMPLINGMODE</a>	This command sets or queries the sampling mode
<a href="#">ACQUIRE:STATE</a>	Starts, stops, or returns acquisition state
<a href="#">ACQUIRE:STOPAFTER</a>	Sets or queries whether the acquisition is continuous or single sequence
<a href="#">FASTACQ?</a>	Enables, disables, or returns state of Fast Acquisition mode
<a href="#">FASTACQ:HIACQRATE</a>	Sets or queries the state of FastAcq optimization for capturing the details with a higher sample rate
<a href="#">FASTACQ:STATE</a>	Returns the Fast Acquisition state

## Alias Command Group

Alias commands allow you to define new commands as a sequence of standard commands. You may 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 2-14: Alias Commands

Command	Description
<a href="#">ALIAS</a>	Sets or queries the alias state
<a href="#">ALIAS:CATALOG?</a>	Returns a list of the currently defined alias labels



Table 2-14: Alias Commands (cont.)

Command	Description
<a href="#">ALias:DEFine</a>	Assigns a sequence of program messages to an alias label
<a href="#">ALias:DELEte</a>	Removes a specified alias
<a href="#">ALias:DELEte:ALL</a>	Deletes all existing aliases
<a href="#">ALias:DELEte:NAME</a>	Removes a specified alias
<a href="#">ALias:STATE</a>	Sets or queries the alias state

## 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 digital signals to be used in the bus
- Specify its display style

---

**NOTE.** *These commands are available only on the MSO70000 series of products.*

---

### Bus Mnemonics

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

Table 2-15: Bus Mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is 1 through 16.

Table 2-16: Bus Commands

Command	Description
<a href="#">BUS:B&lt;x&gt;:DISplay:DECOde:FILE</a>	Sets or queries the symbol table file for the specified bus
<a href="#">BUS:B&lt;x&gt;:DISplay:DECOde:STATe</a>	Sets or queries the format for displaying numeric values in the 'bus' waveform of the specified bus
<a href="#">BUS:B&lt;x&gt;:DISplay:FORMat</a>	Sets or queries the format for displaying numeric values in the 'bus' waveform of the specified bus
<a href="#">BUS:B&lt;x&gt;:DISplay:TYPe</a>	Sets or queries the display style for the specified bus
<a href="#">BUS:B&lt;x&gt;:I2C:CLOCK:SOUrce</a>	Sets or queries the I2C clock (SCLK) source for the specified bus

Table 2-16: Bus Commands (cont.)

Command	Description
BUS:B<x>:I2C:DATa:SOUrce	Sets or queries the I2C data (SDA) source 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	Sets or queries the waveform label for the specified bus
BUS:B<x>:PARAllel:CLOCK:EDGE	Determines which edges of its clock signal cause a clocked parallel bus to sample new states
BUS:B<x>:PARAllel:CLOCK:SOUrce	Sets or queries the Parallel clock source for the specified bus
BUS:B<x>:PARAllel:ISCLOCKED	Determines whether the bus operates in a clocked or asynchronous fashion
BUS:B<x>:PARAllel:SOURCES	Sets or queries the member signals for the Parallel mode of the bus to or reports them as an MSB-to-LSB ordered list
BUS:B<x>:POSition	Sets or queries the position waveform 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:DATa:DELay	Sets or queries the SPI data (DATA) delay 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: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>:TYPE	Sets or queries the bus type specified
SELEct:B<x>	Sets or queries the display state for the specified bus

## Calibration Command Group

The Calibration commands provide information about the current state of instrument calibration and allow you to initiate internal signal path calibration (SPC). Commands that are specific to factory calibration are not described in this manual; however, they are described in the service manual (located on your Product Software CD-ROM in PDF format). You can also order a printed copy (see Recommended Accessories in the online help for the instrument).

**Table 2-17: Calibration Commands**

Command	Description
<a href="#">CALibrate?</a>	Returns the internal and factory calibration status
<a href="#">*CAL?</a>	Instructs the instrument to perform self-calibration and returns the calibration status when complete
<a href="#">CALibrate:CALProbe:CH&lt;x&gt;?</a>	Performs a probe calibration for the selected channel and returns the calibration status
<a href="#">CALibrate:INTERNAL</a>	Starts the internal signal path calibration
<a href="#">CALibrate:INTERNAL:START</a>	Starts the internal signal path calibration
<a href="#">CALibrate:INTERNAL:STATus?</a>	Returns the current status of the internal signal path calibration
<a href="#">CALibrate:PRObestate:CH&lt;x&gt;?</a>	Returns the probe calibration status for the probe of the selected channel
<a href="#">CALibrate:RESults?</a>	Returns the status of all calibration subsystems without performing an SPC operation
<a href="#">CALibrate:RESults:SPC?</a>	Returns the results of the last SPC operation

## 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 cursor 1 and cursor 2, such as waveform source, cursor position, and cursor color.

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).
- **Horizontal Bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them.
- **Waveform Cursors.** Consists of two cursors you can independently assign to a waveform. These cursors provide the same readouts that the vertical and horizontal bar cursors provide. Waveform cursors enable you to conveniently measure waveform amplitude and time. In XY or XYZ 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.** Consists of two pairs of independent horizontal and vertical cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors, depending on the style selected, consist of the intersection of a vertical and horizontal line, an X, or a vertical line with an X. These cursors have no association with any waveform, other than they inherit the color of the waveform they are assigned to.

**Table 2-18: Cursor Commands**

Command	Description
CURSor?	Returns all cursor settings
CURSor:FUNCTioN	Sets or queries the cursor type
CURSor:HBArs?	Returns hbar cursor settings
CURSor:HBArs:DELTA?	Returns hbars cursors vertical difference
CURSor:HBArs:POSITioN<x>	Sets or queries the hbar cursor<x> vertical position
CURSor:HBArs:UNIts?	Returns hbar cursor units
CURSor:LINEStYle	Sets or queries the cursor line style
CURSor:MODE	Sets or queries whether cursors move in unison or separately
CURSor:SCREEN:StYle	Sets or queries the cursor type for screen mode
CURSor:SCREEN:XPOSITioN<x>	Sets or queries the x position of the specified screen cursor
CURSor:SCREEN:YPOSITioN<x>	Sets or queries the y position of the specified screen cursor
CURSor:SOUrce<x>	Sets or queries the source for cursor <x>
CURSor:STATE	Turns cursors on or off or returns their state

Table 2-18: Cursor Commands (cont.)

Command	Description
CURSor:VBArS	Sets or queries the position of vertical bar cursors
CURSor:VBArS:DELtA?	Returns the difference between vbar cursors
CURSor:VBArS:POSItion<x>	Sets or queries the vbar cursor<x> horizontal position
CURSor:VBArS:POS<x>	Sets or queries the horizontal position for vertical bar cursors
CURSor:VBArS:UNItS	Sets or queries the units for vbar cursors
CURSor:WAVEform	Sets or queries the current settings for waveform cursors
CURSor:WAVEform:HDELTA?	Returns the horizontal difference between waveform cursors
CURSor:WAVEform:HPOS<x>?	Returns the position of waveform cursor <x>
CURSor:WAVEform:POSItion<x>	Sets or queries the position of waveform cursor <x>
CURSor:WAVEform:STyle	Sets or queries the cursor type for waveform mode
CURSor:WAVEform:UNItS	Sets or queries the units for waveform cursors
CURSor:WAVEform:VDELTA?	Returns the vertical difference between waveform cursors
CURSor:XY?	Returns the current settings for XY cursors
CURSor:XY:PRODELtA?	Returns the product of the difference between the cursors X positions and Y positions
CURSor:XY:PRODUct<x>?	Returns the product of the X and Y positions for the specified cursor
CURSor:XY:RADIUS<x>?	Returns the radius of the specified cursor
CURSor:XY:RATDELtA?	Returns ratio of the difference between the cursors X position and Y position
CURSor:XY:RAtIO<x>?	Returns ratio of the X (horizontal) and Y (vertical) position for the specified cursor
CURSor:XY:RDELtA?	Returns the Dr value
CURSor:XY:READOUT	Sets or queries the XY cursor readout mode
CURSor:XY:RECTX<x>	Sets or queries the X cursor position in rectangular coordinates
CURSor:XY:RECTY<x>	Sets or queries the Y cursor position in rectangular coordinates
CURSor:XY:THDELtA?	Returns the XY cursor angle delta in polar coordinates

Table 2-18: Cursor Commands (cont.)

Command	Description
CURSor:XY:THETA<x>?	Returns the XY cursor angle in polar coordinates
CURSor:XY:XDELta?	Returns the XY cursor $\Delta X$ value in rectangular coordinates
CURSor:XY:YDELta?	Returns the XY cursor $\Delta Y$ value in rectangular coordinates

## Diagnostics Command Group

The Diagnostic commands control the selection and execution of diagnostic tests.

Table 2-19: Diagnostics Commands

Command	Description
DIAg:CONTROL:HALT	Enables or disables halting on first diagnostic failure
DIAg:CONTROL:LOOP	Enables or disables looping of diagnostics
DIAg:FAILURES:CLEAR	Sets and queries the clearing of pass/fail information from data structures, not the Event Log, at the start of diagnostic tests
DIAg:EXECUTE	Executes currently selected set of diagnostics
DIAg:ITEM?	Returns all data associated with a selected menu item
DIAg:ITEM:FAILURES?	Returns the total number of failures that occurred
DIAg:ITEM:NAME?	Returns the name of the selected menu item
DIAg:ITEM:RESULT?	Returns the results of the last test executed on this item
DIAg:ITEM:SUBITEMS?	Returns the number of subitems associated with this item
DIAg:LEVEL	Sets the current level of diagnostic test hierarchy
DIAg:LOOPS?	Returns the number of times the diagnostics were completed during the last execution
DIAg:NAME?	Returns the subsystem name, area, and test name of the current diagnostic test
DIAg:NAME:AREA?	Returns the selected area of the current diagnostic test
DIAg:NAME:SUBSYS?	Returns the subsystem of the current diagnostic test

Table 2-19: Diagnostics Commands (cont.)

Command	Description
DIAg:NAME:TEST?	Returns the name of the current diagnostic test
DIAg:NUMITEMS?	Returns the number of items on the currently selected level of test hierarchy
DIAg:RESults?	Returns a brief pass or fail status of the last test execution
DIAg:RESults:VERBose?	Returns a more explanatory message about the results of the last diagnostic execution
DIAg:SElect:ALL	Selects all available diagnostics
DIAg:SElect:AREA	Selects one of the available diagnostic areas
DIAg:SElect:LAST	Sets the last item of a group of items from the same level of test hierarchy
DIAg:SElect:SUBSYS	Selects one of the available diagnostic subsystems
DIAg:SElect:TEST	Selects one of the available diagnostic tests
DIAg:STATE	Sets the instrument operating state
DIAg:STOP	Terminates the execution of diagnostics
TEST	Selects and executes an item at any level of the test hierarchy
TEST:RESults?	Returns a brief pass or fail status of the last test execution
TEST:RESults:VERBose?	Returns a more explanatory message about the results of the last test execution
TEST:STOP	Terminates the execution of the test

## Digital Command Group

Use the commands in the Digital Command Group to acquire 24 digital signals and analyze them. The digital signals includes:

- Signals from each of the 16 data channels of its digital probes
- Signals derived from each of the four analog acquisition channels
- Signals derived from each of the four math expressions

**NOTE.** *These commands are available only on the MSO70000 series of products.*

Table 2-20: Digital Commands

Command	Description
CQ<x>:THRESHold	Sets or queries the threshold for converting the specified clock/qualifier signal to digital form
DISplay:DIGital:HEIght	Sets or queries the height of the digital input waveform and the label associated with the channel
D<x>:LABEL	Sets or queries the label that appears for the specified digital input on the display
D<x>:POSition	Sets or queries the position for the specified digital input
D<x>:PROBE:ID:SERnumber?	Queries the serial number of the digital probe that provides the specified digital signal
D<x>:PROBE:ID:TYPE?	Queries the type of digital probe that provides the specified digital signal
D<x>:THRESHold	Sets or queries the threshold level for the digital signal specified
REF<x>:THRESHold	sets or queries the comparable threshold for converting the reference signal to digital form.
SElect:DALL	Sets the display state of all the digital inputs
SElect:DIGTraces:COMbination	Turns on the digital channels that have binary digits as 1
SElect:DIGTraces:LIST	This command turns on the specified digital channels or returns the list of digital channels that are on
SElect:D<x>	Sets or queries the display state for the digital input specified

## Display Control Command Group

Use the commands in the Display Control Command Group to change the graticule style, the displayed intensities, and to set the characteristics of the waveform display.

You can set the display of date and time; cursor, histogram, mask, and measurement readouts; measurement annotations, and the mode in which waveforms are displayed.



There are six color palettes from which you can select:

- **Normal.** Displays hues and lightness levels for best overall viewing.
- **Temp.** Displays areas of the waveform with the highest sample density in warmer colors (red shades) while the areas of lowest sample density appear in cooler colors (blue shades).
- **Spectral.** Displays areas of the waveform with the highest sample density in blue shades while the areas of lowest sample density appear in red shades.
- **Green.** Displays waveforms in shades of green. Areas of the waveform with the highest sample density appear in lighter green shades while the areas of lowest sample density appear in darker green shades.
- **Gray.** Displays waveforms in shades of gray. Areas of the waveform with the highest sample density appear in lighter gray shades while the areas of lowest sample density appear in darker gray shades.
- **User** Allows you to create a customized color palette.

Use the commands to set the style that best displays your waveforms and graticule display properties.

---

**NOTE.** *The mode you choose globally affects all displayed waveforms.*

---

Command	Description
<a href="#">DISplay?</a>	Returns current display settings
<a href="#">DISplay:CLOCK</a>	Sets or queries the display of the date/time stamp
<a href="#">DISplay:COLOr?</a>	Returns color group settings
<a href="#">DISplay:COLOr:MATHCOLOr</a>	Sets or queries the color to be used for math traces
<a href="#">DISplay:COLOr:PALEtte:IMAGEView</a>	Sets or queries the color palette for imageview waveforms
<a href="#">DISplay:COLOr:PALEtte:RECORDView</a>	Sets or queries the color palette for recordview waveforms
<a href="#">DISplay:COLOr:PALEtte:USEr</a>	Returns the user palette group settings
<a href="#">DISplay:COLOr:PALEtte:USEr:CARet</a>	Sets or queries the user caret color
<a href="#">DISplay:COLOr:PALEtte:USEr:CH&lt;x&gt;</a>	Sets or queries the user palette channel colors
<a href="#">DISplay:COLOr:PALEtte:USEr:GRATICule</a>	Sets or queries the user palette graticule
<a href="#">DISplay:COLOr:PALEtte:USEr:HISTogram</a>	Sets or queries the user palette histogram color
<a href="#">DISplay:COLOr:PALEtte:USEr:MASK</a>	Sets or queries the user palette mask color
<a href="#">DISplay:COLOr:PALEtte:USEr:MASKHighlight</a>	Sets or queries the user palette mask hits color

Command	Description
DISplay:COLOr:PALEtte:USEr:MATH<x>	Sets or queries the user palette math colors
DISplay:COLOr:PALEtte:USEr:REF<x>	Sets or queries the user palette reference colors
DISplay:COLOr:PALEtte:USEr:WAVEform	Sets or queries the user palette waveform colors
DISplay:COLOr:REFCOLOr	Sets or queries the color to be used for reference traces
DISplay:FILTer	Sets or queries the type of interpolation to use for the display
DISplay:FORMat	Sets or queries the display format
DISplay:GRAticule	Sets or queries the type of graticule that is displayed
DISplay:INTENSITy?	Returns the waveform and graticule saturation levels
DISplay:INTENSITy:WAVEform:IMAGeView	Sets or queries the waveform saturation level for imageview waveforms
DISplay:INTENSITy:WAVEform:RECORDView	Sets or queries the waveform saturation level for recordview waveforms
DISplay:PERsistence	Sets or queries display persistence setting
DISplay:PERsistence:RESET	Clears the persistence data
DISplay:SCREENExt?	Returns all screen text settings
DISplay:SCREENExt:LABel<x>?	Sets or queries the screen text setting for a given label
DISplay:SCREENExt:LABel<x>:FONTCOLor	Sets or queries the screen text label font color
DISplay:SCREENExt:LABel<x>:FONTNAmE	Sets or queries the screen text label font name for a given label
DISplay:SCREENExt:LABel<x>:FONTSIze	Sets or queries the screen text label font size for a given label
DISplay:SCREENExt:LABel<x>:FONTSYle	Sets or queries the screen text label font style for a given label
DISplay:SCREENExt:LABel<x>:NAMe	Sets the text to be displayed for a given label
DISplay:SCREENExt:LABel<x>:STATe	Sets or queries the screen text label state for a given label
DISplay:SCREENExt:LABel<x>:XPOS	Sets or queries the horizontal position of a given label
DISplay:SCREENExt:LABel<x>:YPOS	Sets or queries the vertical position of a given label
DISplay:SCREENExt:STATe	Sets or queries the state of the display of screen text
DISplay:SHOWREmote	Sets or queries the state of the remote display feature and is equivalent to selecting Display Remote from the Display menu

Command	Description
<a href="#">DISplay:STYle</a>	Sets or queries data display style
<a href="#">DISplay:TRIGBar</a>	Sets or queries the display setting of the trigger level indicator bar(s)
<a href="#">DISplay:TRIGT</a>	Sets or queries the display of the trigger point indicator
<a href="#">DISplay:VARpersist</a>	Sets or queries the persistence decay time
<a href="#">DISplay:WAVEform</a>	Sets or queries the display of waveform traces

## E-mail Command Group

Commands in the E-mail group allow you to send e-mail to one or more designated recipients whenever a selected event, such as a trigger, mask test failure (Option MTM only), or a limit test failure occurs in the instrument.

Using this feature, you do not have to continually monitor the instrument for the event.

When an event occurs, the instrument will send an SMTP mail message to one or more designated recipients through the specified mail server. The message can include any of the following:

- Screen image
- Waveform data
- Measurement data at the time of the event

---

**NOTE.** *Your instrument must be connected to the Local Area Network (LAN) using Transfer C Protocol/Internet Protocol (TCP/IP). The recipient must also have an SMTP mail server. To send e-mail for a mask test failure, you must have Mask Testing (Option MTM) installed.*

---

**Table 2-21: E-mail Commands**

Command	Description
<a href="#">EMail</a>	Sends a test e-mail message or sets the current e-mail sent count to zero
<a href="#">EMail:ATTEMpts</a>	Sets or queries the number of times that an attempt will be made to send e-mail to the SMTP e-mail server
<a href="#">EMail:AUTHLogin</a>	Sets or queries the login name that will be used if the SMTP e-mail server requires one for authentication

Table 2-21: E-mail Commands (cont.)

Command	Description
EEmail:AUTHPassword	Sets the password that will be used if the SMTP e-mail server requires one for authentication
EEmail:COUNt?	Returns the number of e-mails that have been sent since Email on Event was armed
EEmail:FRom	Sets or queries the From line in the e-mail
EEmail:HOSTwanted	Sets or queries the hostname that will be used when e-mail is sent to the SMTP e-mail server
EEmail:IMAGe	Sets or queries whether image data is included in the e-mail
EEmail:MASK	Sets or queries whether e-mail is sent when a mask test failure occurs
EEmail:MAXSize	Sets or queries the maximum size (in megabytes) of e-mail that can be sent to the SMTP server
EEmail:MEASUrement	Sets or queries whether measurement data is included in the e-mail
EEmail:NUMEMails	Sets or queries the number e-mails that can be sent when Email onEvent is armed
EEmail:SMTPPort	Sets or queries the SMTP port number that the e-mail server uses
EEmail:SMTPServer	Sets or queries the address of the SMTP mail server
EEmail:STATUS?	Queries the status of e-mail
EEmail:TIMEOut	Sets or queries the global timeout in seconds
EEmail:TO	Sets or queries the address of the recipient(s) of an e-mail. Multiple addresses are separated with a semicolon (;)
EEmail:TRIGger	Sets or queries whether e-mail is sent when a trigger occurs
EEmail:WAVEform	Sets or queries whether waveform data is included in the e-mail

## Error Detector Command Group

The error detector provides error detection on many serial standards. This section provides error detector theory and a list of commands in the error detector command group.

### Error Detector Theory of Operation

The Tektronix error detector option provides the following:

- Automated receiver testing using AWG7000 Series
- SATA Bit Error Detection: 1.5 Gb/s, 3.0 Gb/s, and 6.0 Gb/s Standards
- SATA Frame Error Detection: 1.5 Gb/s, 3.0 Gb/s, and 6.0 Gb/s Standards
- USB 3.0 Symbol Error Detection: 5 Gb/s Standard
- PCIe Gen1 and Gen2 Bit Error Detection: 5 Gb/s Standard
- Illegal Character and Disparity Error Detection: 1.25 Gb/s to 6.25 Gb/s
- Tracking for Spread Spectrum Clocks with up to 5000 ppm (downspread)

The following table lists the various types of error detection:

**Table 2-22: Types of error detection**

Error Type	(Units) Error Rate	(Units) Resolution	Effectiveness	Limitations
Character	Number of character errors per number of characters tested	Character	90% to 95%	Can miss errors if a character is changed to a legal character.
Frame	Number of frame errors per number of frames tested	Frame	100% (finds all errors)	Many bit errors can equal one frame error
Symbol	Number of symbol errors per number of symbols tested	Symbol	100% (finds all errors)	Several bit errors can equal one symbol error
Bit	Number of bit errors per number of bits tested	Bit	100% (finds all errors)	Pattern length limited by memory depth

For non-USB bit error testing, you must provide the signal test pattern length and sync pattern. Align characters must be provided for all test types. All of these parameter values depend on the actual signal test pattern. The signal test pattern length is generally known from the signal generator setup. The align character is a 10-bit character with both RD+ and RD– disparity values and must actually

be present in the signal test pattern in order to keep the receiver aligned to the 10-bit character boundaries. The sync pattern is a unique 10-, 20-, 30-, or 40-bit sequence (one to four 10-bit characters) that must actually be present in the signal test pattern to align the signal so that bit-by-bit comparisons can be made. The sync pattern does not have to be “at the beginning” of the signal test pattern but it can be anywhere in the signal test pattern, as the fixed-length signal test pattern is sent repetitively by the signal generator.

To better understand the bit error detector, examine and use the setup files that contain the parameter settings in GPIB command format. Signal test patterns are supported with the following setup files:

- Any8B10BCJTPatBitErrorSetup.txt
- Any8B10BCJTPcharacterErrorSetup.txt
- PCIeCJTPatBit ErrorSetup.txt
- SataCJTPatBitErrorSetup.txt
- SataCJTPatCharactErrorSetup.txt
- SataFrameComplianceFrameErrorSetup.txt
- SataHFTPbitErrorSetjup.txt
- SataLBPbitErrorSetup.txt
- SataLFTPbitErrorSetup.txt
- SataMFTPbitErrorSetup.txt
- UsbCP0\_SKPbitErrorSetup.txt
- UsbCP0\_SKPcharacterErrorSetup.txt
- UsbCP0\_SKPsymbolErrorSetup.txt

The oscilloscope error detector has the ability to learn the pattern from the signal test pattern. This capability can be used when you suspect that your DUT may have unintentional design/logic errors that modify the signal test pattern before it reaches the oscilloscope. In this case, the oscilloscope can learn the modified signal test pattern, so that testing can continue even in the presence of the design/logic error.

The USB bit error and USB symbol error detection follow the same concept as SATA frame error detection. In frame error detection, the received serial data is kept aligned to 10-bit boundaries, converted from 10-bit characters to 8-bit characters, monitored for Start of Frame (SOF) and End of Frame (EOF) symbols, scrambled per the specification, to both compute the Cyclic Redundancy Check (CRC) and extract the transmitted CRC from the data. The align primitives can be detected and rejected if desired.

### Example of Programmatic Interface (PI) Commands Used for SATA Frame Error

A 6 Gb/s Lone Bit Pattern (LBP) compliance pattern is injected into oscilloscope Ch1.

Setup serial trigger with 8b10b coding at 6.0 Gb/s for SATA at 6 Gb/s using the following PI commands:

```
*RST //Default Setup puts you into Edge Trigger on Ch1.
```

```
TRIG:A:TYPE SERIAL //Select Serial Trigger
```

```
TRIG:A:SERIAL:SOURCE Ch1 //Select serial Source Ch1.
```

```
TRIG:A:SERIAL:CODE S8B10B //Select 8b10b Coding.
```

```
TRIG:A:SERIAL:STANDARD SATA6_0 //Select Standard (SATA1_5, SATA3_0, SATA6_0).
```

Alternatively, setup serial trigger with 8b10b coding at 6 Gb/s using the user interface.

1. Press **Default Setup** button.
2. Select Serial Trigger from the Trigger drop-down list.
  - a. Select Serial Source Ch1
  - b. Select Coding = S8b10b
  - c. Select SATA6\_0 Standard (or set bitrate = 6 Gb/s).

The following error detector PI commands enable SATA frame error testing on the LBP signal test pattern:

```
TRIG:A:SERIAL:ERROR:ENABLE ON
```

```
TRIG:A:SERIAL:ERROR:FILE:RECALL  
"SataFrameComplianceFrameErrorSetup.txt"
```

```
TRIG:A SETLEVEL
```

```
TRIG:A:SERIAL:ERROR:FRAME:TEST START
```

Send the following error detector PI command to terminate frame error testing and return to serial trigger:

```
TRIG:A:SERIAL:ERROR:ENABLE OFF
```

When frame error detector is used, a count of illegal characters and disparity errors is also provided. The elapsed time in seconds and DD:HH:MM:SS units is available through the PI.

Table 2-23: Error Detector Commands

Command	Description
<b>General Purpose Group</b>	
TRIGger:A:SERIAL:ERRORdetector:DURATION:COUNT	Sets or queries the test duration count as number of bits, frames, symbols, or characters to be tested for error testing
TRIGger:A:SERIAL:ERRORdetector:DURATION:SECONDS	Sets or queries the test duration in seconds for error testing
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME	Sets or queries the test duration time in days, hours, minutes, and seconds for error detector
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:DAYs	Sets or queries the test duration time days component for error testing
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:HOuRS	Sets or queries the test duration time hours component for error testing
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:MINUTEs	Sets or queries the test duration time minutes component for error testing
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:SECONDS	Sets or queries the test duration time seconds component for error testing
TRIGger:A:SERIAL:ERRORdetector:ENABle	Sets or queries the status of the error option. ENABle must be ON to activate the error detector
TRIGger:A:SERIAL:ERRORdetector:ERRORLIMIT	Sets or queries the error limit value to use when STOPWHEN is ERROR
TRIGger:A:SERIAL:ERRORdetector:PATTERNNAME	Sets or queries the pattern name stored in the setup file
TRIGger:A:SERIAL:ERRORdetector:STOPWHEN	Sets or queries the stopping condition
TRIGger:A:SERIAL:ERRORdetector:SCRAMBLED	Sets or queries the status of the data scrambling option
TRIGger:A:SERIAL:ERRORdetector:STANDARD	Sets or queries the standard selection for error testing
TRIGger:A:SERIAL:ERRORdetector:TYPE	Sets or queries the error detector type
<b>Bit Error Detection Group</b>	
TRIGger:A:SERIAL:ERRORdetector:BIT:LENGTH	Sets or queries the signal test pattern length in bits for non-USB bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:SYNCPATtern	Sets or queries the bit sync pattern for non-USB bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST	Arguments START and STOP initiate and terminate bit error testing. Argument CLEAR zeroes bit error test results. Argument LEARN copies the test pattern from the signal



Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:COUNT?	Queries the bit error count for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:DURATION?	Queries the elapsed duration of symbols tested for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:MAXALIGN?	Queries the maximum consecutive SATA align primitives or USB skip order sets for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RATE?	Queries the calculated symbol error rate for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RESULTS?	Queries all the results for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:SECONDS?	Queries the elapsed time in seconds for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS?	Queries all of the bit error test status bits
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:MAX_AP?	Queries the MAX_AP status for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:LOCK?	Queries the LOCK status for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SIGNAL?	Queries the SIGNAL status for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:START?	Queries the START status for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SYNC?	Queries the SYNC status for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME?	Queries the elapsed time (in days, hours, minutes, and seconds) for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:DAYS?	Queries the elapsed time days component for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:HOURS?	Queries the elapsed time hours component for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:MINUTES?	Queries the elapsed time minutes component for bit error testing
TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:SECONDS?	Queries the elapsed time seconds component for bit error testing
TRIGger:A:SERIAL:ERRORdetector:SSC	Sets or queries the status of the spread spectrum clock tracking option
<b>Frame Error Detection Group</b>	
TRIGger:A:SERIAL:ERRORdetector:FRAME?	Queries all frame error settings, status, and results
TRIGger:A:SERIAL:ERRORdetector:FRAME:EOF	Sets or queries the End of Frame for frame error testing

Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector:FRAME:INITIALCRCVALue	Sets or queries the initial CRC value for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:SOF	Sets or queries the Start of Frame for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST	Arguments START and STOP initiate and terminate frame error testing. Argument CLEAR zeroes the frame error results.
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:BADCHARS?	Queries the illegal character count for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:COUNT?	Queries the test error count for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DISPARITY?	Queries the disparity error count for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DURATION?	Queries the elapsed duration (in number of frames) tested by the frame
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:MAXALIGNs?	Queries the maximum consecutive aligns encountered for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RATE?	Queries the calculated frame error rate for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RESULTS?	Queries all the results for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:SECONDS?	Queries the elapsed duration in seconds for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS?	Queries all of the frame error status at once
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:MAX_AP?	Queries the MAX_AP status for the frame error test
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:LOCK?	Queries the LOCK status for the frame error test
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:SIGNAL?	Queries the SIGNAL status for the frame error test
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:START?	Queries the START status for frame error tests
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME?	Queries the elapsed time (in days, hours, minutes, and seconds) for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:DAYS?	Queries the elapsed time days component for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:HOURS?	Queries the elapsed time hours component for frame error testing
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:MINUTES?	Queries the elapsed time minutes component for frame error testing

Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:SECONDS?	Queries the elapsed time seconds component for frame error testing
<b>Symbol Error Detection Group</b>	
TRIGger:A:SERIAL:ERRORdetector:SYMBOL?	Queries all symbol error settings, status, and results
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST	Arguments START and STOP initiate and terminate symbol error testing. Argument CLEAR zeroes the symbol error results
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BADCHARS?	Queries the illegal character count for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITCOUNT?	Queries the bit error count (number of bad bits) for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITDURATION?	Queries the elapsed duration (in units of bits) tested for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITRATE?	Queries the calculated bit error rate for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:COUNT?	Queries the symbol error count for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DISPARITY?	Queries the disparity error count for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DURATION?	Queries the elapsed duration for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:MAXALIGNs?	Queries the maximum consecutive skip order sets encountered for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RATE?	Queries the calculated symbol error rate for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RESULTS?	Queries all of the results for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:SECONDS?	Queries the elapsed duration time in seconds for symbol error testing
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS?	Queries all of the status for the symbol error tests
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:MAX_AP?	Queries the MAX_AP status for the symbol error test
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:LOCK?	Queries the LOCK status for the symbol error test
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:SIGNAL?	Queries the SIGNAL status for the symbol error test
TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:START?	Queries the START status for symbol error testing

Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector: SYMBOL:TEST:TIME?	Queries the elapsed time (in days, hours, minutes, and seconds) for symbol error testing
TRIGger:A:SERIAL:ERRORdetector: SYMBOL:TEST:TIME:DAY?	Queries the elapsed time days component for symbol error testing
TRIGger:A:SERIAL:ERRORdetector: SYMBOL:TEST:TIME:HOUR?	Queries the elapsed time hours component for symbol error testing
TRIGger:A:SERIAL:ERRORdetector: SYMBOL:TEST:TIME:MINUTE?	Queries the elapsed time minutes component for symbol error testing
TRIGger:A:SERIAL:ERRORdetector: SYMBOL:TEST:TIME:SECOND?	Queries the elapsed time seconds component for symbol error testing
<b>Character Error Detection Group</b>	
TRIGger:A:SERIAL:ERRORdetector: CHARACTER?	Queries all character error settings, status, and results
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST	Argument START and STOP initiate and terminate character error testing. Argument CLEAR zeroes the character error results
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:COUNT?	Queries the number of illegal characters in the signal for error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:DISPCOUNT?	Queries the number of disparity errors in the signal for character error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:DISPRATE?	Queries the calculated disparity error rate for character error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:DURATION?	Queries the elapsed duration in units of characters tested
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:MAXALIGN?	Queries the maximum consecutive aligns encountered for character error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:RATE?	Queries the calculated illegal character error rate for error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:RESULTS?	Queries all of the character error test results
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:SECONDS?	Queries the elapsed duration time seconds component for character error testing
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:STATUS?	Queries all of the status for character error tests
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:STATUS:MAX_AP?	Queries the MAX_AP status for the character error test
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:STATUS:LOCK?	Queries the LOCK status for the character error test
TRIGger:A:SERIAL:ERRORdetector: CHARACTER:TEST:STATUS:SIGNAL?	Queries the SIGNAL status for the character error test

Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:START?	Queries the START status for the character error testing
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME?	Queries the elapsed time (in days, hours, minutes, and seconds) for character error testing
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:DAYs?	Queries the elapsed time days component for character error testing
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:HOuRS?	Queries the elapsed time hours component for character error testing
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:MINuTES?	Queries the elapsed time minutes component for character error testing
TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:SECOndS?	Queries the elapsed time seconds component for character error testing
<b>File Group</b>	
TRIGger:A:SERIAL:ERRORdetector:FILE:RECALL	Initiates a file recall of error detector parameters from a text file
TRIGger:A:SERIAL:ERRORdetector:FILE:SAVE	Initiates a file save of error detector parameters to a text file
<b>Align Pattern Group</b>	
TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter?	Queries both of the align character values
TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDMINUS	Sets or queries the RD- align character value
TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDPLUS	Sets or queries the RD+ align character value
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive?	Queries all of the align primitive values
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS?	Queries the RD- align primitive enable and string values
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS:ENABLE	Sets or queries the status of the RD- align primitive option
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS:PATtern	Sets or queries the RD- align primitive pattern
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS?	Queries the RD+ align primitive enable and string values
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:ENABLE	Sets or queries the status of the RD+ align primitive option
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:PATtern	Sets or queries the RD+ align primitive pattern
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:S8BPATtern	Sets or queries the 32-bit version for align primitive pattern

Table 2-23: Error Detector Commands (cont.)

Command	Description
TRIGger:A:SERIAL:ERRORdetector:MAXALIGNs	Sets or queries the maximum consecutive align primitives before an error
<b>Skip Order Set Group</b>	
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS?	Queries the RD- skip order set enable and string values
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:ENABLE	Sets or queries the status of the RD- skip order set option
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:PATtern	Sets or queries the RD- skip order set pattern
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS?	Queries the RD+ skip order set enable and string values
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:ENABLE	Sets or queries the status of the RD+ skip order set option
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:PATtern	Sets or queries the RD+ skip order set pattern
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:S8BPATtern	Sets or queries the 32-bit skip order set pattern

## 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 default 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 MSDOS format: [DRIVE:][\PATH\]filename
- Path separators may be either forward slashes (/) or back slashes (\)

---

**NOTE.** Using back slash as a path separator may produce some unexpected results, depending on how your controller application treats escaped characters. Many applications recognize the sequence of 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” may be interpreted as a newline character; “\t” may 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 may 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 2-24: 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 GPIB commands
FILESystem:DELEte	Deletes a named file or directory
FILESystem:DIR?	Returns a list of directory contents
FILESystem:MKDir	Makes a new directory
FILESystem:PRInt	Prints a named file to the named port
FILESystem:READFile	Copies the named file to the GPIB port
FILESystem:REName	Assigns a new name to an existing file
FILESystem:RMDir	Deletes the named directory
FILESystem:WRITEFile	Copies the GPIB port block data to a named file

## Hard Copy Command Group

Hard Copy commands enable you to make hard copies of data files or send the data to a specified file. In addition, these commands provide you with information about (and the ability to set) file path settings.

**Table 2-25: Hard Copy Commands**

<b>Command</b>	<b>Description</b>
<a href="#">EXPort</a>	Copies a waveform to a specified file or returns formatting and file information
<a href="#">EXPort:FILEName</a>	Sets or queries the export file path
<a href="#">EXPort:FORMat</a>	Sets or queries the export image format
<a href="#">EXPort:PALEtte</a>	Sets or queries the export color palette
<a href="#">EXPort:READOuts</a>	Sets or queries the position of the readouts for export
<a href="#">EXPort:VIEW</a>	Sets or queries the export screen capture area
<a href="#">HARDCopy</a>	Sends a screen copy to the selected port or returns the selected port and file path
<a href="#">HARDCopy:FILEName</a>	Sets or queries the hard copy file path
<a href="#">HARDCopy:LAYout</a>	Sets or queries the page orientation for hard copy
<a href="#">HARDCopy:PALEtte</a>	Sets or queries the hard copy color palette
<a href="#">HARDCopy:PORT</a>	Sets or queries whether the hard copy data will be sent to a file or printed on the next hard copy command
<a href="#">HARDCopy:READOuts</a>	Sets or queries the position of the readouts for hardcopy
<a href="#">HARDCopy:VIEW</a>	Sets or queries the area of the screen to be hardcopied

## Histogram Command Group

Histogram commands let you select the type of histogram, what part of the waveform should go into the histogram, and histogram statistics. Use commands from this group to do the following:

- Create a histogram of vertical or horizontal values for a channel, math, or reference waveform
- Adjust the limits of the waveform area from which the histogram data is obtained
- Clear histogram count and restart
- Retrieve the histogram data



Table 2-26: Histogram Commands

Command	Description
<a href="#">HIStogram?</a>	Returns all histogram parameters
<a href="#">HIStogram:BOX</a>	Sets or queries the left, top, right, and bottom positions of the histogram box, in source waveform coordinates
<a href="#">HIStogram:BOXPcnt</a>	Sets or queries same as HIStogram:BOX, but in percentage coordinates, with 0,0 upper left and 100,100 lower right
<a href="#">HIStogram:COUNT</a>	Clears histogram count source data and restarts counting
<a href="#">HIStogram:DATA?</a>	Returns a comma-separated list of histogram data numbers
<a href="#">HIStogram:DISPlay</a>	Sets or queries whether histogram data is displayed and the current display type setting
<a href="#">HIStogram:FUNcTion</a>	Sets the type of histogram to create Or queries the current histogram type
<a href="#">HIStogram:MODe</a>	Sets type of histogram to create or turns histograms off Or queries the current histogram type or that histogram display is disabled
<a href="#">HIStogram:SIze</a>	Sets or queries the width (or height) of the histogram on the screen in divisions
<a href="#">HIStogram:SOUrcE</a>	Sets or queries which source waveform will be compared against the histogram box when the histogram testing is enabled
<a href="#">HIStogram:STATE</a>	Sets or queries whether histogram calculations are enabled

## Horizontal Command Group

Horizontal commands control the time bases 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

You may substitute SECdiv for SCALE in the horizontal commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-27: Horizontal Commands

Command	Description
HORizontal?	Returns all learnable settings for the horizontal commands
HORizontal:ACQDURATION?	Returns the time base duration
HORizontal:ACQLENGTH?	Returns the record length
HORizontal:DIGital:RECOrdlength?	Queries the record length of the digital acquisition
HORizontal:DIGital:SAMPLERate?	Queries the sample rate of the digital acquisition
HORizontal:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed
HORizontal:FASTframe?	Returns all settings for the horizontal FastFrame commands
HORizontal:FASTframe:COUNt	Sets or queries the FastFrame frame count
HORizontal:FASTframe:MAXFRames?	Returns the maximum number of FastFrame frames which can be acquired at the current frame length
HORizontal:FASTframe:MULTipleframes:FRAMESart:<wfm>	Sets or queries the start frame number on the specified waveform for FastFrame multiple frames
HORizontal:FASTframe:MULTipleframes:MODE	Sets or queries the mode for FastFrame multiple frames
HORizontal:FASTframe:MULTipleframes:NUMFRames:<wfm>	Sets or queries the number of frames on the specified waveform for FastFrame multiple frames
HORizontal:FASTframe:REF:FRame	Sets or queries the FastFrame reference frame number
HORizontal:FASTframe:REF:SOUrce	Sets or queries the FastFrame Reference waveform source
HORizontal:FASTframe:SELECTED:CH<x>	Sets or queries the FastFrame selected frame number on the specified waveform
HORizontal:FASTframe:SELECTED:MATH<x>	Sets or queries the FastFrame selected frame number on the specified waveform
HORizontal:FASTframe:SELECTED:REF<x>	Sets or queries the FastFrame selected frame number on the specified waveform
HORizontal:FASTframe:SELECTED:SOUrce	Sets or queries the FastFrame source waveform
HORizontal:FASTframe:SEQuence	Sets or queries the FastFrame source waveform
HORizontal:FASTframe:STATE	Sets or queries the state of FastFrame acquisition
HORizontal:FASTframe:SUMFrame	Sets or queries the summary frame mode

Table 2-27: Horizontal Commands (cont.)

Command	Description
HORizontal:FASTframe:TIMESTAMP:ALL:<wfm>?	Returns the frame number and time stamp for each frame between requested frames, inclusive, within the specified waveform
HORizontal:FASTframe:TIMESTAMP:BETWEEN:<wfm>?	Returns the relative trigger for the delta time between the specified frames, within the specified waveform
HORizontal:FASTframe:TIMESTAMP:DELTA:<wfm>?	Returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform
HORizontal:FASTframe:TIMESTAMP:FRAME:<wfm>?	Returns the absolute trigger date and time for the specified frame and waveform
HORizontal:FASTframe:TIMESTAMP:REF?	Returns the absolute trigger date and time for the FastFrame specified reference waveform
HORizontal:FASTframe:TIMESTAMP:SELECTED:<wfm>?	Returns the absolute trigger date and time for the FastFrame Selected, within the specified waveform
HORizontal:FASTframe:TRACK	Sets up or returns the state of FastFrame tracking
HORizontal:FASTframe:XZERO:ALL:CH<x>?	Returns the time from the trigger to the start of the specified frames on the specified channel
HORizontal:FASTframe:XZERO:ALL:REF<x>?	Returns the time from the trigger to the start of the specified frames on the specified reference
HORizontal:FASTframe:XZERO:FRAME:CH<x>?	Returns the time from the trigger to the start of the specified frame on the specified channel
HORizontal:FASTframe:XZERO:FRAME:REF<x>?	Returns the time from the trigger to the start of the specified frame on the specified reference
HORizontal:FASTframe:XZERO:REF?	Sets or queries the time from the trigger to the trigger sample on the reference waveform
HORizontal:FASTframe:XZERO:SELECTED:CH<x>?	Sets or queries the time from the trigger to the trigger sample on the selected channel
HORizontal:FASTframe:XZERO:SELECTED:REF<x>?	Sets or queries the time from the trigger to the trigger sample on the selected reference waveform
HORizontal:MAIN?	Returns the time per division of the main time base
HORizontal[:MAIN]:DELAY:MODE	Sets or queries the main time base trigger delay mode

Table 2-27: Horizontal Commands (cont.)

Command	Description
<code>HORizontal:MAIn:DElay:POSition</code>	Sets or queries the main time base position when Horizontal Delay Mode is turned off
<code>HORizontal:MAIn:DElay:TIME</code>	Sets or queries the main time base trigger delay time
<code>HORizontal:MAIn:INTERPRatio?</code>	Returns the main horizontal time base interpolation ratio
<code>HORizontal:MAIn:POSition</code>	Sets or queries the waveform horizontal position on the display
<code>HORizontal:MAIn:UNIts</code>	Sets or queries the units for the horizontal time base
<code>HORizontal:MAIn:UNIts:STRing</code>	Sets or queries the units string for the horizontal main time base trigger delay
<code>HORizontal:MODE</code>	Sets or queries the horizontal mode. Auto mode is the factory default
<code>HORizontal:MODE:AUTO:LIMITrecordlen</code>	Sets or queries the record length limit used by the auto horizontal mode
<code>HORizontal:MODE:RECOrdlength</code>	Sets or queries the record length
<code>HORizontal:MODE:SAMPLERate</code>	Sets or queries the sample rate
<code>HORizontal:MODE:SCALE</code>	Sets or queries the horizontal scale
<code>HORizontal:POSition</code>	Either positions the waveform horizontally on the display or returns its horizontal position on the display
<code>HORizontal:ROLL</code>	Sets or queries the horizontal roll mode status
<code>HORizontal:TIMEStamp:CH&lt;x&gt;?</code>	Returns the absolute trigger date and time for the specified frame and waveform
<code>HORizontal:TIMEStamp:REF&lt;x&gt;?</code>	Returns the absolute trigger date and time for FastFrame reference

## Limit Test Command Group

Use the commands in the Limit Test Command Group to perform limit testing by selecting channel, math, or reference waveforms and comparing them against a template waveform.

Table 2-28: Limit Test Commands

Command	Description
LIMit?	Returns all settings for the Limit commands
LIMit:BEEP	Causes the instrument to beep when the waveform data exceeds the limits set in the limit test
LIMit:COMpare	Resets the limit test comparison template
LIMit:COMpare:CH<x>	Sets or queries the template against which to compare the waveform acquired from the channel specified by x
LIMit:COMpare:MATH<x>	Sets or queries the template against which to compare the math waveform specified by x
LIMit:COMpare:REF<x>	Sets or queries the template against which to compare the reference waveform specified by x
LIMit:EMail	Sets or queries whether an E-mail is generated when the source waveform data exceeds the limits specified for the limit test
LIMit:HARDCopy	Sets or queries whether a hard copy operation is executed on the waveform when any waveform data exceeds the limit set in the limit test. LIMit:STATE must be set to ON for the hard copy operation to execute
LIMit:HIGHLIGHTHits	Sets or queries whether violation highlighting occurs when limit testing is active, and, if the RESET argument is set, clears the highlighting
LIMit:HIGHLIGHTHits:RESet	Resets the hits highlighting for limit testing
LIMit:LOCK	Sets or queries whether vertical scaling and positioning affect both source and template for template comparison pairs
LIMit:LOG	Sets or queries whether a log file is saved when the source waveform data exceeds the test limits
LIMit:SAVEWFM	Sets or queries whether the source waveform is saved when the source waveform data exceeds the test limits
LIMit:SAVEWFM:FILEName	Sets or queries the path where waveforms or log files will be saved when the waveform data exceeds the limits set by the limit test
LIMit:SRQ	Sets or queries whether a Service Request Interrupt (SRQ) is generated when the waveform data falls outside of the test limits
LIMit:STATE	Sets limit testing on or off or queries whether limit testing is in effect

Table 2-28: Limit Test Commands (cont.)

Command	Description
<a href="#">LIMit:STATus?</a>	Queries the state of limit testing
<a href="#">LIMit:STOPOnviolation</a>	Sets or queries whether acquisitions are stopped when the waveform data exceeds the test limits
<a href="#">LIMit:TEMPlate:STORe</a>	Saves the specified source waveform to the specified reference or file name
<a href="#">LIMit:TEMPlate:TOLerance:HORizontal</a>	Sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform
<a href="#">LIMit:TEMPlate:TOLerance:VERTical</a>	Sets or queries the amount, in units of vertical divisions, by which the source waveform is varied vertically when creating the destination waveform
<a href="#">LIMit:TEMPlate:DESTination</a>	Sets or queries destination reference waveform that the limit template
<a href="#">LIMit:TEMPlate:SOURce</a>	Sets or queries the channel, math waveform, or reference waveform that the <a href="#">LIMit:TEMPlate:STORe</a> command will use

## Low Speed Serial Trigger Command Group

DPO7000 Series only: Use the commands in the Low Speed Serial Trigger Command Group to control serial triggering for the instrument. CAN trigger commands are only available with the CAN option.

Table 2-29: Low Speed Serial Trigger Commands

Command	Description
<a href="#">TRIGger:A:CAN:CONDition</a>	Sets or queries the CAN condition
<a href="#">TRIGger:A:CAN:DATA:DIRection</a>	Sets or queries the CAN trigger condition to be valid on a READ, WRITE, or either
<a href="#">TRIGger:A:CAN:FORMat</a>	Sets or queries the CAN data format
<a href="#">TRIGger:A:CAN:DATA:LEVel</a>	Sets or queries the CAN Trigger threshold for the CAN data source
<a href="#">TRIGger:A:CAN:DATA:SOURce</a>	Sets or queries the CAN data source
<a href="#">TRIGger:A:CAN:DATA:VALue</a>	Sets or queries the binary data string used for CAN Trigger if the trigger condition is ID or IDANDDATA
<a href="#">TRIGger:A:CAN:FRAMEtype</a>	Sets or queries the CAN trigger frame type
<a href="#">TRIGger:A:CAN:IDENTifier:MODE</a>	Sets or queries the CAN trigger identifier mode

Table 2-29: Low Speed Serial Trigger Commands (cont.)

Command	Description
TRIGger:A:CAN:IDENTifier:VALue	Sets or queries the binary address string used for the CAN trigger if the trigger condition is ID or IDANDDATA
TRIGger:A:CAN:PROBE	Sets or queries the probing method used to probe the CAN signal
TRIGger:A:CAN:SPEEd	Sets or queries the bit rate of the CAN system
TRIGger:A:I2C:ADDRess:MODE	Sets or queries the I2C address mode to 7 or 10 bit
TRIGger:A:I2C:ADDRess:RWINClude	Sets or queries the I2C address mode to include the read/write bit as an additional 8th bit of the 7-bit address
TRIGger:A:I2C:ADDRess:TYPE	Sets or queries the I2C address type
TRIGger:A:I2C:ADDRess:VALue	Sets or queries the binary address string used for the I2C trigger if the trigger condition is ADDR or ADDRANDDATA
TRIGger:A:I2C:CLOCK:LEVel	Sets or queries the clock source for the I2C Serial Trigger
TRIGger:A:I2C:CLOCK:SOUrce	Sets or queries the clock source for the I2C Serial Trigger
TRIGger:A:I2C:CONDition	Sets or queries the trigger condition for the I2C trigger
TRIGger:A:I2C:DATA:DIRection	Sets or queries the I2C trigger condition valid on a READ, WRITE, or either
TRIGger:A:I2C:DATA:LEVel	Sets or queries the threshold level for the I2C data source
TRIGger:A:I2C:DATA:SOUrce	Sets or queries the data source for the I2C serial trigger
TRIGger:A:I2C:DATA:STARt	Sets or queries the data start byte
TRIGger:A:I2C:DATA:VALue	Sets or queries the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA
TRIGger:A:I2C:FORMat	Sets or queries the display format for the I2C data value
TRIGger:A:RS232:BAUd	Sets or queries the baud rate for RS232 triggering
TRIGger:A:RS232:DATA:LEVel	Sets or queries the threshold for the RS232 data source
TRIGger:A:RS232:DATA:SOUrce	Sets or queries the RS232 data source
TRIGger:A:RS232:DATA:VALue	Sets or queries the binary data string used for RS232 triggering

Table 2-29: Low Speed Serial Trigger Commands (cont.)

Command	Description
TRIGger:A:RS232:FORMat	Sets or queries the display format for the RS232 data value
TRIGger:A:RS232:PARity	Sets or queries the state of the RS232 parity bit
TRIGger:A:SERIAL:BITRate	Sets or queries the clock/data bit rate
TRIGger:A:SERIAL:CLOCK:LEVel	Sets or queries the serial trigger clock level
TRIGger:A:SERIAL:CLOCK:POLarity	Sets or queries the serial clock polarity
TRIGger:A:SERIAL:CLOCK:SOUrce	Sets or queries the serial data source channel
TRIGger:A:SERIAL:CODe	Sets or queries the signal code
TRIGger:A:SERIAL:DATa:FORMat	Sets or queries how the Pattern string is formatted
TRIGger:A:SERIAL:DATa:PATtern	Sets or queries the data pattern to allow up to 64 bit serial patterns
TRIGger:A:SERIAL:DATa:PATtern:NRZ	Sets or queries the data pattern to allow up to 64 bit serial patterns
TRIGger:A:SERIAL:DATa:PATtern:S8B10B	Sets or queries the data pattern to allow up to 64 bit serial patterns
TRIGger:A:SERIAL:LOCKLen	Sets or queries the length in bits of the repeating bit pattern for pattern lock trigger
TRIGger:A:SERIAL:LOCKOffset	Sets or queries the current bit offset into the pattern lock trigger bit pattern
TRIGger:A:SERIAL:SOUrce	Sets or queries the serial data source channel
TRIGger:A:SERIAL:STANdard	Sets or queries the standard that identifies the code and bit rate
TRIGger:A:SERIAL:TRIGgeron	Sets or queries the trigger on a designated arbitrary bit pattern or lock on a repeating pattern of known length
TRIGger:A:SPI:CONDition	Sets or queries the trigger condition for SPI triggering
TRIGger:A:SPI:DATa:MISO:ACTIVE	Sets or queries the SPI MISO polarity
TRIGger:A:SPI:DATa:MISO:LEVel	Sets or queries the threshold for the SPI MISO data source
TRIGger:A:SPI:DATa:MISO:SOUrce	Sets or queries the MISO data source for the SPI trigger
TRIGger:A:SPI:DATa:MISO:VALue	Sets or queries the binary data string used for the SPI trigger if the trigger condition is set to MISO or MISOMOSI
TRIGger:A:SPI:DATa:MOSI:ACTIVE	Sets or queries the SPI MOSI polarity



Table 2-29: Low Speed Serial Trigger Commands (cont.)

Command	Description
TRIGger:A:SPI:DATA:MOSI:LEVel	Sets or queries the threshold for the SPI MOSI data source
TRIGger:A:SPI:DATA:MOSI:SOUrce	Sets or queries the MOSI data source for the SPI trigger
TRIGger:A:SPI:DATA:MOSI:VALue	Sets or queries the binary data string used for the SPI trigger if the trigger condition is set to MOSI or MISOMOSI
TRIGger:A:SPI:DATA:STARt	Sets or queries the nth data byte on the data source after the signal on the enable slave source switches to the polarity specified by the Slave Select Polarity
TRIGger:A:SPI:FORMat	Sets or queries the SPI trigger data format
TRIGger:A:SPI:SCLK:ACTIVE	Sets or queries the SPI SCLK polarity
TRIGger:A:SPI:SCLK:LEVel	Sets or queries the threshold for the SPI trigger SCLK
TRIGger:A:SPI:SCLK:SOUrce	Sets or queries the SPI SCLK source
TRIGger:A:SPI:SS:ACTIVE	Sets or queries the SPI trigger Slave Select (SS) polarity
TRIGger:A:SPI:SS:LEVel	Sets or queries the threshold for the SPI trigger Slave Select (SS) signal
TRIGger:A:SPI:SS:SOUrce	Sets or queries the SPI trigger Slave Select (SS) source.
<b>NOTE.</b> DPO7000 Series only	

## Mask Command Group

Use the commands in the Mask Command Group to compare incoming waveforms against standard or user-defined telecommunications masks. Standard masks are only available when option MTM is installed. Without option MTM, you need to define your masks. When the instrument detects waveforms that fall inside or outside of the mask limits, it can perform tasks such as generating hard copy, saving the waveform, emitting a beep or an SRQ, stopping and waiting for your input, sending an E-mail message with specified data, logging the time/date of the acquired waveform, or some combination of these actions.

A mask is a set of polygonal regions on the screen. Every vertical line of the screen intersects the polygon in zero, one, or two places, but never in more than two places. (A vertical line that intersects a vertical mask border is counted.) You have to break up more complicated polygons into two separate segments. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. A telecommunications standard requires up to eight of these

segments. Pulse standards always have two segments. Standards with eye patterns usually have three masks, but some have four.

You use the commands in the Mask Command Group to do the following:

- Specify the waveform source to test and the mask to use.
- Specify whether to use, and the size of, mask margins, which allow you to shrink or expand an existing set of polygons by a specified percentage.
- Specify whether to display a readout of hits and the mask on screen. Options also exist for auto fitting the incoming waveforms to match the mask you choose.
- Select industry-standard masks that support a variety of electrical and optical communication standards (available only with option MTM).
- Define and edit your own custom mask; create an entirely new mask, or use a standard mask (option MTM only) as a starting reference, and edit it to meet your needs.
- Enable, disable, or reset the mask counts. Once you turn on mask counting, it remains on until you explicitly turn it off.

Option MTM also provides communication trigger commands that you can use in conjunction with the mask commands, such as TRIGger:A:TYPE Communication. Refer to the Trigger Command Group.

**Table 2-30: Mask Commands**

Command	Description
<a href="#">MASK?</a>	Returns the states of all settable mask parameters
<a href="#">MASK:AUTOAdjust</a>	Optimizes signal position within the mask to minimize hits
<a href="#">MASK:AUTOAdjust:HDELTA</a>	Sets or queries how far autofit searches horizontally
<a href="#">MASK:AUTOAdjust:VDELTA</a>	Sets or queries how far autofit searches vertically
<a href="#">MASK:AUTOSet:AUTOAdjust</a>	Controls what happens at the end of a mask autose. If ON, the horz/vert auto adjustment is run ONCE to potentially improve the result of mask autose
<a href="#">MASK:AUTOSet:HPOS</a>	Controls whether the algorithm will attempt to change the horizontal position of the signal while attempting to autose
<a href="#">MASK:AUTOSet:HSCALE</a>	Controls whether the algorithm will attempt to change the horizontal scale while attempting to autose

Table 2-30: Mask Commands (cont.)

Command	Description
<a href="#">MASK:AUTOSet:MODE</a>	Controls whether an autoset will be automatically done after a standard mask is selected
<a href="#">MASK:AUTOSet:OFFSETAdj</a>	Sets autoset not to enforce the rule that, for pulse standards, requires 0 V to be in a certain place in the mask. Instead, autoset will attempt to measure the DC offset in the signal and use scope offset or position controls to compensate for the DC offset in the signal
<a href="#">MASK:AUTOSet:STANdard</a>	Selects the standard mask in autoset
<a href="#">MASK:AUTOSet:TRIGger</a>	Controls if a MASK:AUTOSet affects the trigger level
<a href="#">MASK:AUTOSet:USER:ONE</a>	Sets or queries the level 1 used by autoset for user masks
<a href="#">MASK:AUTOSet:USER:TYPe</a>	Sets or queries how autoset scales a mask
<a href="#">MASK:AUTOSet:USER:ZERo</a>	Sets or queries the level 0 used by autoset for user masks
<a href="#">MASK:AUTOSet:VPOS</a>	Controls whether the autoset algorithm will affect the vertical position (or offset) of the signal while attempting to autoset
<a href="#">MASK:AUTOSet:VSCAlE</a>	Controls whether the autoset algorithm will attempt to change the vertical scale while attempting to autoset
<a href="#">MASK:COpy:USER</a>	Copies the currently displayed mask to the User mask
<a href="#">MASK:COUNT</a>	Resets to zero the number of hits for all mask segments on the specified source
<a href="#">MASK:COUNT:FAILURES?</a>	Returns the number of pass/fail tests that have failed
<a href="#">MASK:COUNT:HITS?</a>	Returns the total number of mask hits on all mask segments
<a href="#">MASK:COUNT:SEG&lt;m&gt;:HITS?</a>	Returns the number of hits in mask segment <m>
<a href="#">MASK:COUNT:STATE</a>	Sets or queries the mask hits count state
<a href="#">MASK:COUNT:TESTS?</a>	Returns the number of pass/fail tests that have occurred
<a href="#">MASK:COUNT:TOTal?</a>	Returns the sum of all hits in all mask segments
<a href="#">MASK:COUNT:VIOLATIONS?</a>	Returns the number of test violations that have occurred in the current pass/fail test

Table 2-30: Mask Commands (cont.)

Command	Description
<a href="#">MASK:COUNT:WAVEFORMS?</a>	Returns the number of waveforms that have been acquired and processed during pass/fail testing
<a href="#">MASK:DISplay</a>	Sets or queries whether the defined masks are displayed on the screen
<a href="#">MASK:FILTer</a>	Sets or queries whether the selected optical reference receiver mask filter is enabled
<a href="#">MASK:HIGHLIGHTHits</a>	Sets or queries whether the hits in a mask are highlighted in different colors than other waveform data
<a href="#">MASK:INVert</a>	Controls whether the mask is drawn inverted
<a href="#">MASK:LOCK</a>	Sets or queries the mask lock to waveform mode
<a href="#">MASK:MARgin:PERCent</a>	Sets or queries the mask margin percentage
<a href="#">MASK:MARgin:STATE</a>	Sets or queries the state of the mask margins
<a href="#">MASK:MASKPRE:AMPLitude</a>	Sets or queries the current mask's nominal pulse amplitude in volts
<a href="#">MASK:MASKPRE:HSCAle</a>	Sets or queries the nominal timing resolution used to draw the current mask, in time/division
<a href="#">MASK:MASKPRE:HTRIGPOS</a>	Sets or queries the nominal trigger position (pulse leading edge) used to draw the current mask as a fraction of the display width
<a href="#">MASK:MASKPRE:PATTERNBITS</a>	Sets or queries the number of bits used for serial trigger for the current mask standard
<a href="#">MASK:MASKPRE:PRESAMPBITS</a>	Sets or queries the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing
<a href="#">MASK:MASKPRE:RECOrdlength</a>	Sets or queries the nominal record length for pulse mask testing
<a href="#">MASK:MASKPRE:TRIGTOSAMP</a>	Sets or queries the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
<a href="#">MASK:MASKPRE:VOFFSet</a>	Sets or queries the nominal vertical offset, in volts, used to vertically offset the input channels
<a href="#">MASK:MASKPRE:VPOS</a>	Sets or queries the nominal vertical position, in divisions, used to vertically position the input channels
<a href="#">MASK:MASKPRE:VSCAle</a>	Sets or queries the nominal vertical scale, in volts per division, used to vertically scale the input channels

Table 2-30: Mask Commands (cont.)

Command	Description
MASK:MASKPRE:WIDTH	Sets or queries the nominal bit width in seconds
MASK:POLarity	Sets or queries the input waveform polarity for the pass/fail test
MASK:SEG<m>	Deletes the specified mask segment from the Current mask
MASK:SEG<m>:NR_Pt?	Returns the number of points that make up the specified mask segment of the Current mask
MASK:SEG<m>:POINTS	Sets or queries the X-Y user coordinates of all points in the Current mask segment
MASK:SOUrce	Sets or reports which source will be compared against the mask(s) when counting is turned on
MASK:STANdard	Deletes the existing mask and sets the selected standard mask
MASK:STOPOnviolation	Sets or queries the stop on violation setting
MASK:TEST:AUX:COMPLetion	Sets or queries the test:aux on pass/fail test completion mode
MASK:TEST:AUX:FAILure	Sets or queries the test:aux status on pass/fail test failure mode
MASK:TEST:BEEP:COMPLetion	Sets or queries the beep on pass/fail test completion mode
MASK:TEST:BEEP:FAILure	Sets or queries the beep status on pass/fail test failure mode
MASK:TEST:DELay	Sets or queries the amount of time, in seconds, to wait after starting pass/fail testing to evaluate waveforms
MASK:TEST:HARDCopy	Sets or queries the hard copy in pass/fail failure mode
MASK:TEST:LOG:FAILure	Sets or queries the log status on pass/fail test failure mode
MASK:TEST:REPeat	Sets or queries the repeat pass/fail test on completion mode
MASK:TEST:SAMple	Sets or queries the minimum number of waveform database (WfmDB) points the instrument can acquire before it stops a single sequence acquisition or stops running a mask test
MASK:TEST:SAMple:THReshold	Sets or queries the minimum number of hits in mask regions needed to cause the pass/fail status to change from PASSING to FAILING

Table 2-30: Mask Commands (cont.)

Command	Description
MASK:TEST:SAVEWFM	Sets or queries the save waveform to file on failure setting
MASK:TEST:SAVEWFM:FILENAME	Sets or queries the name of the directory and file to use with the MASK:TEST:SAVEWFM command
MASK:TEST:SRQ:COMPLETION	Sets or queries an SRQ on pass/fail test completion mode
MASK:TEST:SRQ:FAILURE	Sets or queries the SRQ status on pass/fail test failure mode
MASK:TEST:STATE	Sets or queries the state of mask pass/fail testing
MASK:TEST:STATUS?	Returns the pass/fail test status
MASK:TEST:STOP:FAILURE	Sets or queries the stop status on pass/fail test failure mode
MASK:TEST:THRESHOLD	Sets or queries the number of failed tested waveforms needed in each pass/fail test to cause the pass/fail status to change from PASSING to FAILING
MASK:TEST:WAVEFORM	Sets or queries the number of waveforms to test during pass/fail testing
MASK:USER:AMPLITUDE	Sets or queries the current mask's nominal pulse amplitude in volts
MASK:USER:BITRATE	Sets or queries the user mask's bit rate
MASK:USER:HSCALE	Sets or queries the nominal timing resolution used to draw the mask, in time/division
MASK:USER:HTRIGPOS	Sets or queries the nominal trigger position (pulse leading edge) used to draw the mask as a fraction of the display width
MASK:USER:LABEL	Sets or queries the label assigned to a user mask
MASK:USER:PATTERNBITS	Sets or queries the number of bits used for serial trigger for the User mask standard
MASK:USER:PRESAMPBITS	Sets or queries the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing
MASK:USER:RECORDLENGTH	Sets or queries the nominal record length for pulse mask testing
MASK:USER:SEG<m>	Deletes the specified mask segment from the User mask
MASK:USER:SEG<m>:NR_Pt?	Returns the number of points that make up the specified User mask segment

Table 2-30: Mask Commands (cont.)

Command	Description
<code>MASK:USER:SEG&lt;m&gt;:POINTS</code>	Sets or queries the X-Y user coordinates of all points in the specified User mask segment
<code>MASK:USER:TRIGTOSAMP</code>	Sets or queries the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
<code>MASK:USER:VOFFSet</code>	Sets or queries the nominal vertical offset, in volts, used to vertically offset the input channels
<code>MASK:USER:VPOS</code>	Sets or queries the nominal vertical position, in divisions, used to vertically position the input channels
<code>MASK:USER:VSCALE</code>	Sets or queries the nominal vertical scale, in volts per division, used to vertically scale the input channels
<code>MASK:USER:WIDTH</code>	Sets or queries the nominal bit width in seconds
<code>RECALL:MASK</code>	Recalls a stored mask as the User mask
<code>SAVE:MASK</code>	Saves the User mask to a mass storage file

## Math Command Group

Use the commands in the Math Command Group to create and define math waveforms. You can define and display up to four math waveforms simultaneously on four channel models and up to two on two channel models. 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 update or 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.

**Table 2-31: Math Commands**

<b>Command</b>	<b>Description</b>
<code>MATH&lt;x&gt;?</code>	Returns the specified math waveform settings
<code>MATH&lt;x&gt;:DEFine</code>	Sets or queries the math<x> waveform definition for the specified waveform
<code>MATH&lt;x&gt;:FILTer:MODE</code>	Sets or queries the filter rise time parameter (centered or shifted)
<code>MATH&lt;x&gt;:FILTer:RISetime</code>	Sets or queries the filter rise time parameter
<code>MATH&lt;x&gt;:LABel:NAME</code>	Sets or queries the label string used for annotating the displayed math waveform
<code>MATH&lt;x&gt;:LABel:XPOS</code>	Sets or queries the X screen offset where the math waveform label is displayed
<code>MATH&lt;x&gt;:LABel:YPOS</code>	Sets or queries the Y screen offset where the math waveform label is displayed
<code>MATH&lt;x&gt;:NUMAVg</code>	Sets or queries the acquisition number at which the averaging algorithm will begin exponential averaging
<code>MATH&lt;x&gt;:SPECTral?</code>	Returns the specified math waveform spectral setups
<code>MATH&lt;x&gt;:SPECTral:CENTER</code>	Sets or queries the center frequency of the spectral analyzer output data span
<code>MATH&lt;x&gt;:SPECTral:GATEPOS</code>	Sets or queries the position of the center of the gate
<code>MATH&lt;x&gt;:SPECTral:GATEWIDTH</code>	Sets or queries the gate width input to the spectral analyzer
<code>MATH&lt;x&gt;:SPECTral:LOCK</code>	Sets or queries the state of the spectral locking for the specified math waveform
<code>MATH&lt;x&gt;:SPECTral:MAG</code>	Sets or queries the units of the SpectralMag function in the specified math string
<code>MATH&lt;x&gt;:SPECTral:PHASE</code>	Sets or queries the units of the SpectralPhase function in the specified math string
<code>MATH&lt;x&gt;:SPECTral:REFLevel</code>	Sets or queries the vertical position of the output data from the spectral analyzer
<code>MATH&lt;x&gt;:SPECTral:REFLEVELOffset</code>	Sets or queries the spectral reference level offset used for calculating the dB value
<code>MATH&lt;x&gt;:SPECTral:RESBw</code>	Sets or queries the resolution bandwidth of the spectral analyzer



Table 2-31: Math Commands (cont.)

Command	Description
<code>MATH&lt;x&gt;:SPECTral:SPAN</code>	Sets the top of the frequency span to a value that is closest to the specified value Or queries the specified math waveform's current frequency span value
<code>MATH&lt;x&gt;:SPECTral:SUPPress</code>	Sets or queries the phase suppression threshold
<code>MATH&lt;x&gt;:SPECTral:UNWRap</code>	Enables or disables or returns the state of the phase unwrap of the spectral analyzer output data
<code>MATH&lt;x&gt;:SPECTral:WINDow</code>	Sets or queries the window function used to multiply the input data to the spectral analyzer
<code>MATH&lt;x&gt;:UNITString</code>	Sets or queries the units used for the math waveform; command overrides the default unit string
<code>MATH&lt;x&gt;:VERTical:POSition</code>	Sets or queries the specified math waveform vertical position
<code>MATH&lt;x&gt;:VERTical:SCALE</code>	Sets or queries the specified math waveform vertical scale (per div)
<code>MATHArbfit&lt;x&gt;:FILEpath</code>	Sets the file path for a file of filter coefficients and reads the file
<code>MATHArbfit&lt;x&gt;:READFile</code>	Forces a reread of the filter file for the specified arbitrary filter
<code>MATHVAR?</code>	Sets or queries all 8 numerical values you can use within math expressions
<code>MATHVAR:VAR&lt;x&gt;</code>	Sets or queries a numerical value you can use within math expressions
<code>MATH&lt;x&gt;:THRESHold</code>	Sets or queries the comparable threshold for converting the math signal to digital form

## Measurement Command Group

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

Up to eight automated measurements can be displayed on the screen. In the commands, these measurement readouts are named MEAS<x>, where <x> can be 1 through 8.

In addition to the eight displayed measurements, the measurement commands let you specify an additional measurement, IMMEd. The immediate measurement has no front panel equivalent. Immediate measurements are never displayed.

Because they are computed only when needed, immediate measurements slow the waveform update rate less than displayed measurements.

Whether you use displayed or immediate measurements, use the VALue query to obtain measurement results.

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

**Table 2-32: Measurement Commands**

Command	Description
MEASUrement?	Returns all measurement parameters
MEASUrement:ANNOtation:IMMEDStAte	Sets or queries the state of immediate measurement annotation
MEASUrement:ANNOtation:NUMX?	Returns the number of horizontal measurement annotations being displayed
MEASUrement:ANNOtation:NUMY?	Returns the number of vertical measurement annotations being displayed
MEASUrement:ANNOtation:StAte	Sets or queries the state of visible measurement annotations
MEASUrement:ANNOtation:TYPe	Sets or queries the type of annotations being used
MEASUrement:ANNOtation:X<x>?	Returns the value of the specified annotation
MEASUrement:ANNOtation:Y<x>?	Returns the value of the specified annotation
MEASUrement:GATing	Sets or queries the measurement gating
MEASUrement:IMMed?	Returns all immediate measurement setup parameters
MEASUrement:IMMed:DELay?	Returns information about the immediate delay measurement
MEASUrement:IMMed:DELay:DIREction	Sets or queries the search direction to use for immediate delay measurements
MEASUrement:IMMed:DELay:EDGE[1]	Sets or queries the slope of the edge used for immediate delay “from” waveform measurements
MEASUrement:IMMed:DELay:EDGE2	Sets or queries the slope of the edge used for immediate delay “to” waveform measurements
MEASUrement:IMMed:MEtHod	Specifies or queries the method used to calculate the 0% and 100% reference level for immediate measurements
MEASUrement:IMMed:NOISe	Sets or queries the location of the noise measurement on the eye waveform
MEASUrement:IMMed:REFLevel?	Returns the reference levels

Table 2-32: Measurement Commands (cont.)

Command	Description
MEASUrement:IMMed:REFLevel:ABSolute: HIGH	Sets or queries the high reference level
MEASUrement:IMMed:REFLevel:ABSolute: LOW	Sets or queries the low reference level
MEASUrement:IMMed:REFLevel:ABSolute: MID<x>	Sets or queries the middle reference levels
MEASUrement:IMMed:REFLevel:METHod	Sets or queries the method used to calculate measurements
MEASUrement:IMMed:REFLevel:PERCent: HIGH	Sets or queries the top level for rise time (90% level)
MEASUrement:IMMed:REFLevel:PERCent: LOW	Sets or queries the low level for rise time (10% level)
MEASUrement:IMMed:REFLevel:PERCent: MID<x>	Sets or queries the mid levels for measurements
MEASUrement:IMMed:SOUrce<x>	Sets or queries the source to measure “to” or “from” for phase or delay immediate measurements
MEASUrement:IMMed:SOUrce<x>:SIGType	Sets or queries the type of input signal used for the specified measurement source
MEASUrement:IMMed:TYPe	Sets or queries the type of the immediate measurement
MEASUrement:IMMed:UNIts?	Returns the units of the immediate measurement
MEASUrement:IMMed:VALue?	Returns the value of the immediate measurement
MEASUrement:MEAS<x>?	Returns all measurement parameters for the displayed measurement
MEASUrement:MEAS<x>:COUNt?	Returns the number of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:DELay?	Returns the delay measurement parameters for specified measurement
MEASUrement:MEAS<x>:DELay:DIREction	Sets or queries the search direction to use for delay measurements
MEASUrement:MEAS<x>:DELay:EDGE<x>	Sets or queries the slope of the edge to use for delay “from” or “to” waveform measurements
MEASUrement:MEAS<x>:MAXimum?	Returns the maximum value found since the last statistical reset
MEASUrement:MEAS<x>:MEAN?	Returns the mean value accumulated since the last statistical reset
MEASUrement:MEAS<x>:METHod	Sets or queries the method used to calculate the 0% and 100% reference level

Table 2-32: Measurement Commands (cont.)

Command	Description
MEASUrement:MEAS<x>:MINImum?	Returns the minimum value found since the last statistical reset
MEASUrement:MEAS<x>:NOISe	Sets or queries the location of the noise measurement on the eye waveform for MEAS<x>
MEASUrement:MEAS<x>:REFLevel?	Returns the reference levels
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH	Sets or queries the top level for rise time (90% level)
MEASUrement:MEAS<x>:REFLevel:ABSolute:LOW	Sets or queries the low level for rise time (10%)
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID<x>	Sets or queries the mid1 or mid2 level for measurements
MEASUrement:MEAS<x>:REFLevel:METhod	Sets or queries the reference level units used for measurement calculations
MEASUrement:MEAS<x>:REFLevel:PERCent:HIGH	Sets or queries the top level for rise time (90% level)
MEASUrement:MEAS<x>:REFLevel:PERCent:LOW	Sets or queries the low level for rise time (10% level)
MEASUrement:MEAS<x>:REFLevel:PERCent:MID<x>	Sets or queries the mid1 or mid2 level for delay measurements
MEASUrement:MEAS<x>:SOUrce<x>	Sets or queries the source to measure “from” or “to” for phase or delay measurements
MEASUrement:MEAS<x>:SOUrce<x>:SIGType	Sets or queries the type of input signal used for the specified measurement source for the specified measurement slot
MEASUrement:MEAS<x>:STATE	Sets or queries whether the specified measurement slot is computed and displayed
MEASUrement:MEAS<x>:STDdev?	Returns the standard deviation of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:TYPE	Sets or queries the measurement<x> type
MEASUrement:MEAS<x>:UNIts?	Returns measurement<x> units
MEASUrement:MEAS<x>:VALue?	Returns the value of measurement<x>
MEASUrement:METhod	Sets or queries the method used for calculating reference levels
MEASUrement:NOISe	Sets or queries the location of the noise measurement on the eye waveform
MEASUrement:REFLevel:ABSolute:HIGH	Sets or queries the top reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW	Sets or queries the low reference level for rise time

Table 2-32: Measurement Commands (cont.)

Command	Description
MEASUrement:REFLevel:ABSolute:MID<x>	Sets or queries the mid1 or mid2 reference level for measurements
MEASUrement:REFLevel:MEtHod	Sets or queries the method for assigning high and low reference levels
MEASUrement:REFLevel:PERCent:HIGH	Sets or queries the top reference percent level for rise time
MEASUrement:REFLevel:PERCent:LOW	Sets or queries the low reference percent level for rise time
MEASUrement:REFLevel:PERCent:MID<x>	Sets or queries the mid1 or mid 2 reference percent level for waveform measurements
MEASUrement:SOUrce<x>:SIGType	Sets or queries the type of input signal used for the specified measurement source
MEASUrement:STATIstics:COUNt	Clears existing measurement statistics from memory
MEASUrement:STATIstics:MODE	Turns measurement statistics on or off and sets or queries which pair of statistical accumulations is displayed
MEASUrement:STATIstics:WEIghting	Sets or queries the 'time constant' for mean and standard deviation statistical accumulations

## Miscellaneous Command Group

Miscellaneous commands do not fit into other categories.

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

Table 2-33: Miscellaneous Commands

Command	Description
AUXout?	Returns the auxiliary out setup
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
AUTOSet	Sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform

Table 2-33: Miscellaneous Commands (cont.)

Command	Description
BELI	In previous models, this command sounded the audio indicator. Provided for backward compatibility
CMDBatch	Turns command batching on or off Or returns the state of command batching
DATE	Sets or queries the date that the instrument can display
*DDT	Sets or queries the commands that will be executed by the group execute trigger
HEADer	Sets or queries the Response Header Enable State
HDR	Sets or queries the Response Header Enable State
ID?	Returns identifying information about the instrument and its firmware
*IDN?	Returns the instrument identification code
LOCK	Sets or queries the front panel lock state
*LRN?	Returns a listing of instrument settings
NEWpass	Changes the password for user protected data
PASSWord	Provides access for changing user protected data
REM	Specifies a comment which is ignored by the instrument
ROSc:OUT:FREQuency	Sets or queries the selected frequency for the timebase reference output signal
ROSc:SOUrce	Sets or queries the selected source for the time base reference oscillator
ROSc:STATE?	Returns whether the time base reference oscillator is locked
ROSc:TRACking	Sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source
SET?	Returns a listing of instrument settings
TEKSecure	Initializes both waveform and setup memories
TIME	Sets or queries the time displayed by the instrument
*TRG	Performs the group execute trigger (GET)
*TST?	Tests the GPIB interface and returns status

Table 2-33: Miscellaneous Commands (cont.)

Command	Description
UNLock	Unlocks front panel
VERBose	Sets or queries the verbose state

## 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 2-34: Save and Recall Commands

Command	Description
*RCL	Restores the state of the instrument from a copy of the setting stored in memory
*SAV	Stores the current instrument state to a specified memory location
*SDS	Changes the specified setup to reference the factory setup instead of the user setup
DELEte:SETUp	Removes stored setups from memory and initializes the location with the factory default setups
DELEte:WAVEform	Deletes (one or all of) the stored reference waveforms from memory
FACTory	Resets the instrument to factory default settings
RECAll:SETUp	Recalls saved instrument settings
RECAll:SETUp:DESKew	Sets or queries the deskew values that are affected by a default setup or a recalled setup
RECAll:WAVEform	Recalls a stored waveform to a reference location
SAVe:SETUp	Saves the current front panel setup to a specified memory location or file
SAVe:WAVEform	Saves a waveform to one of four reference memory locations or a mass storage file
SAVe:WAVEform:FILEFormat	Sets or queries the format for saved waveforms
SETUp:NAMe	Sets or queries the user-defined setup label

## Search and Mark Command Group

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

**Table 2-35: Search and Mark Commands**

Command	Description
<a href="#">MARK</a>	Moves to the next or previous reference mark, or queries the current mark setup
<a href="#">MARK:CREATE</a>	Creates a mark on a specified waveform or all waveforms in a column
<a href="#">MARK:DELEte</a>	Deletes a mark on a particular waveform, all waveforms in a column, the selected mark, or all marks
<a href="#">MARK:FREE?</a>	Returns how many marks are available for use
<a href="#">MARK:SELECTED:END?</a>	Returns the end of the selected mark, 0 to 100% of the waveform
<a href="#">MARK:SELECTED:FOCUS?</a>	Returns the focus of the selected mark, 0 to 100% of the waveform
<a href="#">MARK:SELECTED:LABel</a>	Set or queries the label for the selected mark
<a href="#">MARK:SELECTED:MARKSINCOLumn?</a>	Returns the number of marks in the current zoom pixel column
<a href="#">MARK:SELECTED:OWNer?</a>	Returns the owner of the selected mark
<a href="#">MARK:SELECTED:SOUrce?</a>	Sets or queries the source waveform for the selected mark
<a href="#">MARK:SELECTED:STARt?</a>	Returns the starting point of the selected mark, 0 to 100% of the waveform
<a href="#">MARK:SELECTED:STATE</a>	Returns the on or off state of the selected mark. The selected mark is at or near the center of the display
<a href="#">MARK:SELECTED:ZOOm:POSition?</a>	Returns the position of the selected mark, 0 to 100% of the zoom overview window
<a href="#">MARK:SELECTED:ZOOm:SCAle?</a>	Returns the scale of the selected mark, 0 to 100% of the zoom overview window
<a href="#">MARK:TOTal?</a>	Returns how many marks are currently in use
<a href="#">SEARCH:SEARCH&lt;x&gt;</a>	Converts the specified search marks to the user marks. <x> can be 1 to 8
<a href="#">SEARCH:SEARCH&lt;x&gt;:COPy</a>	Copies the search criteria to the trigger, the trigger criteria to a search, or copy search to search. <x> is the search number
<a href="#">SEARCH:SEARCH&lt;x&gt;:STATE</a>	Selects which search is selected



Table 2-35: Search and Mark Commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TOTAL?	Returns the total number of matches for the search. The total number of matches may be more than the number of marks placed
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOBitrate	Sets or queries the auto bit rate setting
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOReflevel	Sets or queries the auto reference levels setting
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:BITRate	Sets or queries the DDR memory bit rate
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSActive	Sets or queries whether the expected chip-select will be an active HIGH or LOW
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSLevel	Sets or queries the DDR memory search chip-select level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSMode	Sets or queries the chip-select mode
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSSource	Sets or queries the chip select source
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:DATASource	Sets or queries the DDR memory data source
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:HYSterisis	Sets or queries the hysteresis percent for DDR memory searches
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:HIGH	Sets or queries the value of the data HIGH reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:LOW	Sets or queries the value of the data LOW reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:MID	Sets or queries the value of the data MID reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:HIGH	Sets or queries the value of the strobe HIGH reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:LOW	Sets or queries the value of the strobe LOW reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:MID	Sets or queries the value of the strobe MID reference level
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STANdard	Sets or queries the DDR standard type for a DDR Memory analysis search
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STROBESource	Sets or queries the strobe source for a DDR Memory analysis search
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:CH<x>	Sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the channel number

Table 2-35: Search and Mark Commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:EDGE: SLOpe:MATH<x>	Sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the math waveform number
SEARCH:SEARCH<x>:TRIGger:A:EDGE: SLOpe:REF<x>	Sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the reference waveform number
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. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:GLItch: POLarity:CH<x>	Sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:GLItch: POLarity:MATH<x>	Sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:GLItch: POLarity:REF<x>	Sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:GLItch: TRIGIF	Sets or queries the accept or reject state for a glitch trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:GLItch: WIDTH	Sets or queries the glitch width for a glitch trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:LEVel: CH<x>	Sets or queries the level for an edge trigger search to determine where to place a mark. <x> is the search number or the channel number. Each channel can have an independent level
SEARCH:SEARCH<x>:TRIGger:A:LEVel: MATH<x>	Sets or queries the math waveform level for an edge trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:LEVel: REF<x>	Sets or queries the specified reference waveform level for a trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number

Table 2-35: Search and Mark Commands (cont.)

Command	Description
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. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CH<x>	Sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the channel number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:MATH<x>	Sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the math waveform number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:REF<x>	Sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the reference channel number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>	Sets or queries the channel threshold level for a pattern or state trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH<x>	Sets or queries the math waveform threshold level for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the Math waveform number
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x>	Sets or queries the reference waveform threshold level for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn	Sets or queries the pattern trigger pattern test that determines where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:LESSLimit	Sets or queries the time the pattern trigger less limit test uses to determines where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit	Sets or queries the time the pattern trigger more limit test uses to determines where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:PULse:SOUrce	Sets or queries the source waveform for a pulse trigger search to determine where to place a mark. <x> is the search number. This is the source for Glitch, Width, Runt, Timeout or Window searches

**Table 2-35: Search and Mark Commands (cont.)**

<b>Command</b>	<b>Description</b>
SEARCH:SEARCH<x>:TRIGger:A:RUNT: POLarity:CH<x>	Sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: POLarity:MATH<x>	Sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: POLarity:REF<x>	Sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:HIGH:CH<x>	Sets or queries the channel threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the channel number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:HIGH:MATH<x>	Sets or queries the math waveform threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the math waveform number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:HIGH:REF<x>	Sets or queries the reference waveform threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:LOW:CH<x>	Sets or queries the channel threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the channel number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:LOW:MATH<x>	Sets or queries the math waveform threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the math waveform number
SEARCH:SEARCH<x>:TRIGger:A:RUNT: THReshold:LOW:REF<x>	Sets or queries the reference waveform threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the reference waveform number
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. <x> is the search number

Table 2-35: Search and Mark Commands (cont.)

Command	Description
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. <x> is the search number
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. <x> is the search number
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. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:CH<x>	Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:MATH<x>	Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:REF<x>	Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:SOUrce	Sets or queries the data source setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold:CH<x>	Sets or queries the data threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold:MATH<x>	Sets or queries the data threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATa:THReshold:REF<x>	Sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number
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. <x> is the search number
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. <x> is the search number

**Table 2-35: Search and Mark Commands (cont.)**

<b>Command</b>	<b>Description</b>
SEARCH:SEARCH<x>:TRIGger:A:STATE: CLOCK:EDGE	Sets or queries the clock slope setting for a state trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:STATE: CLOCK:SOUrce	Sets or queries the clock source of the twelve waveforms for a state trigger search to determine where to place a mark. <x> is the search number. The hardware trigger is always on the channel CH4
SEARCH:SEARCH<x>:TRIGger:A:STATE: WHEn	Sets or queries the state trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut: POLarity:CH<x>	Sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut: POLarity:MATH<x>	Sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut: POLarity:REF<x>	Sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or reference waveform number
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. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:DELTATime	Sets or queries the transition time setting for a transition trigger search to determine where to place a mark.<x> is the search number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:POLarity:CH<x>	Sets or queries the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:POLarity:MATH<x>	Sets or queries the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:POLarity:REF<x>	Sets or queries the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number

Table 2-35: Search and Mark Commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:HIGh:CH<x>	Sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:HIGh:MATH<x>	Sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:HIGh:REF<x>	Sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:LOW:CH<x>	Sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:LOW:MATH<x>	Sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:THReshold:LOW:REF<x>	Sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A: TRANSition:WHEn	Sets or queries the condition setting for a transition trigger search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:TYPE	Sets or queries the trigger type setting for a search to determine where to place a mark. <x> is the search number
	<b>NOTE.</b> <i>Except EDGE all other options must be purchased.</i>
SEARCH:SEARCH<x>:TRIGger:A:WIDTH: HIGHLimit	Sets or queries the upper time limit setting for a width search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:WIDTH: LOWLimit	Sets or queries the lower time limit setting for a width search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:WIDTH: POLarity:CH<x>	Sets or queries the polarity for a width search to determine where to place a mark. <x> is the search number or channel number

**Table 2-35: Search and Mark Commands (cont.)**

<b>Command</b>	<b>Description</b>
SEARCH:SEARCH<x>:TRIGger:A:WIDth: POLarity:MATH<x>	Sets or queries the polarity for a width search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:WIDth: POLarity:REF<x>	Sets or queries the polarity for a width search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:WIDth: WHEn	Sets or queries the width search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: EVENT	Sets or queries the window search to determine where to place a mark. <x> is the search number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:HIGH:CH<x>	Sets or queries the channel waveform threshold HIGH level for a window search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:HIGH:MATH<x>	Sets or queries the channel waveform threshold HIGH level for a window search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:HIGH:REF<x>	Sets or queries the channel waveform threshold HIGH level for a window search to determine where to place a mark. <x> is the search number or reference number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:LOW:CH<x>	Sets or queries the channel waveform threshold LOW level for a window search to determine where to place a mark. <x> is the search number or channel number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:LOW:MATH<x>	Sets or queries the channel waveform threshold LOW level for a window search to determine where to place a mark. <x> is the search number or math waveform number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: THReshold:LOW:REF<x>	Sets or queries the channel waveform threshold LOW level for a window search to determine where to place a mark. <x> is the search number or reference waveform number
SEARCH:SEARCH<x>:TRIGger:A:WINDow: WIDth	Sets or queries the width setting for a window search to determine where to place a mark. <x> is the search number
SEARCH:STOP	Sets or queries the stop of acquisitions when a search finds an event



Table 2-35: Search and Mark Commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MARgin	Sets or queries the termination logic margin for back-to-back writes
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MAXCas	Sets or queries the maximum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MINCas	Sets or queries the minimum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse

## 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 on the GPIB bus. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (\*) character.

Table 2-36: Status and Error Commands

Command	Description
*CLS	Clears status
*ESE	Sets or queries the bits in the Event Status Enable Register
*ESR?	Returns the contents of the Standard Event Status Register
*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 list of options installed in the instrument
*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

Table 2-36: Status and Error Commands (cont.)

Command	Description
*STB?	Returns the contents of the Status Byte Register
*WAI	Prevents the instrument from executing further commands until all pending operations finish
ALLEv?	Returns all events and their messages
BUSY?	Returns instrument status
DESE	Sets or queries the bits in the Device Event Status Enable Register
EVENT?	Returns event code from the event queue
EVMsg?	Returns event code, message from the event queue
EVQty?	Returns the number of events in the event queue

## TekLink Command Group

Use the commands in the TekLink command Group to set up TekLink and MultiScope triggering.

Table 2-37: TekLink Commands

Command	Description
TEKLink:CONNECTION?	Returns the current TekLink network connection
TEKLink:REFClk	Sets or queries the current state of the instrument's TekLink Reference Output
TRIGger:MULTIscope	Sets or queries the state of MultiScope triggering, either ENABLE or DISable
TRIGger:MULTIscope:ALIGN:COMPLETED?	Returns whether the oscilloscope has successfully completed the MultiScope trigger align procedure
TRIGger:MULTIscope:ALIGN:DESKEW?	Sets or queries the deskew time, in nanoseconds, for horizontal positioning of the waveform
TRIGger:MULTIscope:ALIGN:MRTTime	Sends or returns the maximum round trip time to each instrument in a calibration cycle
TRIGger:MULTIscope:ALIGN:SETSTATE	Measures the round trip time between sending a trigger out the TekLink port and receiving it back from the port
TRIGger:MULTIscope:ALIGN:VALue?	Queries the delay time, in nanoseconds, for trigger out of the TekLink trigger out line

Table 2-37: TekLink Commands (cont.)

Command	Description
TRIGger:MULTiScope:ALIGN	Starts the TekLink trigger alignment procedure on all oscilloscopes which are participating with the oscilloscope that is in MultiScope triggering
TRIGger:MULTiScope:DELAy?	Returns the delay time for trigger out of the TekLink trigger out line
TRIGger:MULTiScope:LOGic	Sets or queries the role of the instrument in the MultiScope trigger network
	Sets or queries the trigger type for the TekLink trigger
TRIGger:MULTiScope:ROLe	Sets or queries the role of the instrument in the MultiScope trigger network, either PRODUCER or CONSUMER

## Trigger Command Group

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

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

You can set the A or B 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 or B triggers to pulse or logic modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. Video triggering enables you to trigger on the most common Standard Definition, High Definition, and custom video standards. You can trigger on all fields, all lines, or individual video lines. Video trigger commands are only available on DPO7000 instruments.

If you have Option MTM (Mask Testing) installed, you also have access to a communication mode. When this mode is selected, several communication trigger commands are available that work in conjunction with the mask commands. Refer to Mask Command Group.

Table 2-38: Trigger Commands

Command	Description
TRIGger	Forces a trigger event to occur or returns current trigger parameters for the instrument
TRIGger:A	Sets A trigger level to 50% or returns current A trigger parameters
TRIGger:{A B}:EDGE:COUPling	Sets or queries the type of coupling for the edge trigger
TRIGger:{A B}:EDGE:SLOpe	Sets or queries the slope for the edge trigger
TRIGger:{A B}:EDGE:SLOpe:AUX	Sets or queries the slope for the edge trigger for the AUX In input
TRIGger:{A B}:EDGE:SLOpe:CH<x>	Sets or queries the slope for the edge trigger for the channel
TRIGger:{A B}:EDGE:SOUrce	Sets or queries the source for the edge trigger
TRIGger:{A B}:LEVel	Sets or queries the level for the trigger
TRIGger:{A B}:LEVel:CH<x>	Sets or queries the level for the trigger for the channel
TRIGger:{A B}:EDGE?	Returns the source, coupling and source for the edge trigger
TRIGger:{A B}:EDGE:COUPling:CH<x>	Sets or queries the type of coupling for the A or B trigger for the specified channel
TRIGger:{A B}:LOGic?	Returns all of the A trigger logic parameters
TRIGger:{A B}:LOGic:CLAss	Sets or queries the type of trigger logic
TRIGger:{A B}:LOGic:FUNcTION	Sets or queries the logical combination of the input channels for the logic trigger
TRIGger:{A B}:LOGic:INPut?	Returns the logic trigger input expected for channel 1, 2, and 3
TRIGger:{A B}:LOGic:INPut:ALL	Sets or queries the A or B logic trigger input condition for all the channels
TRIGger:{A B}:LOGic:INPut:CH<x>	Sets or queries the A or B logical trigger input for the specified channel
TRIGger:{A B}:LOGic:INPut:FORMat	Sets or queries the A or B logic trigger pattern format
TRIGger:{A B}:LOGic:PATtern:INPut:CH<x>	Sets or queries the logic pattern trigger input for the indicated channel
TRIGger:{A B}:LOGic:PATtern:WHEn	Sets or queries the condition for generating the logic pattern trigger
TRIGger:{A B}:LOGic:PATtern:WHEn:LESSLimit	Sets or queries the maximum time that the selected pattern may be true and still generate a logic pattern trigger

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:{A B}:LOGic:PATtern:WHEn:MORELimit	Sets or queries the minimum time that the selected pattern may be true and still generate a logic pattern trigger
TRIGger:{A B}:LOGic:PATtern?	Returns the conditions for generating a logic pattern trigger
TRIGger:{A B}:LOGic:SETHold:CLOCK:EDGE	Sets or queries the clock edge polarity for setup and hold triggering
TRIGger:{A B}:LOGic:SETHold:CLOCK:SOURce	Sets or queries the clock source for the logic trigger setup and hold input
TRIGger:{A B}:LOGic:SETHold:CLOCK:THReshold	Sets or queries the clock voltage threshold for setup and hold trigger
TRIGger:{A B}:LOGic:SETHold:CLOCK:THReshold:CH<x>	Sets or queries the clock voltage threshold for setup and hold trigger for the channel
TRIGger:{A B}:LOGic:SETHold:CLOCK?	Returns clock edge polarity, voltage threshold and source input for setup/hold triggering
TRIGger:{A B}:LOGic:SETHold:DATA:SOURce	Sets or queries the data source for the setup and hold trigger
TRIGger:{A B}:LOGic:SETHold:DATA:THReshold	Sets or queries the data voltage threshold for setup and hold trigger
TRIGger:{A B}:LOGic:SETHold:DATA:THReshold:CH<x>	Sets or queries the data voltage threshold for setup and hold trigger for the channel
TRIGger:{A B}:LOGic:SETHold:DATA?	Returns the voltage threshold and data source for the setup/hold trigger
TRIGger:{A B}:LOGic:SETHold:HOLDTime	Sets or queries the hold time for the setup and hold violation triggering
TRIGger:{A B}:LOGic:SETHold:QUALify	Sets or queries the setup and hold violation trigger qualifications
TRIGger:{A B}:LOGic:SETHold:SETTime	Sets or queries the setup time for setup and hold violation triggering
TRIGger:{A B}:LOGic:SETHold?	Returns clock edge polarity, voltage threshold and source; data voltage threshold and source; and setup/hold times for setup and hold violation triggering
TRIGger:{A B}:LOGic:STATE:CLOCK:SOURce	Sets or queries the A or B logic trigger state of clock source
TRIGger:{A B}:LOGic:STATE:INPut:CH<x>	Sets or queries the slope for indicated channel of the logic state trigger
TRIGger:{A B}:LOGic:STATE:WHEn	Sets or queries the condition for generating a logic state trigger
TRIGger:{A B}:LOGic:STATE?	Returns the data input and trigger criteria for the logic state trigger

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:{A B}:LOGic:THReshold:CH<x>	Sets or queries the logic state trigger threshold voltage for the specified channel
TRIGger:{A B}:LOWerthreshold:CH<x>	Sets or queries CH<x> lower trigger level for TRIGger: LVLsrcpreference
TRIGger:{A B}:LOGic:THReshold?	Returns the threshold voltage for all channels in logic trigger
TRIGger:{A B}:PULse?	Returns the pulse trigger parameters
TRIGger:{A B}:PULse:CLAss	Sets or queries the type of pulse on which to trigger
TRIGger:{A B}:PULse:GLItch:POLarity	Sets or queries the polarity for the pulse glitch trigger
TRIGger:{A B}:PULse:GLItch:POLarity:CH<x>	Sets or queries the polarity for the A or B pulse glitch trigger for the channel
TRIGger:{A B}:PULse:GLItch:QUALify	Sets or queries the glitch trigger qualifications
TRIGger:{A B}:PULse:GLItch:TRIGIF	Sets or queries the acceptance or rejection of pulse glitch trigger, based on width
TRIGger:{A B}:PULse:GLItch:WIDTH	Sets or queries the width of the A pulse glitch trigger
TRIGger:{A B}:PULse:GLItch?	Returns the current A glitch pulse trigger parameters
TRIGger:{A B}:PULse:RUNT:POLarity	Sets or queries the polarity for the pulse runt trigger
TRIGger:{A B}:PULse:RUNT:POLarity:CH<x>	Sets or queries the polarity for the A or B pulse runt trigger for the channel
TRIGger:{A B}:PULse:RUNT:QUALify	Sets or queries the polarity for the A or B pulse runt trigger
TRIGger:{A B}:PULse:RUNT:THReshold:BOTh	Sets the upper and lower switching thresholds for the A pulse runt trigger
TRIGger:{A B}:PULse:RUNT:THReshold:HIGH	Sets or queries the upper limit for the A pulse runt trigger
TRIGger:{A B}:PULse:RUNT:THReshold:LOW	Sets or queries the lower limit for the A pulse runt trigger
TRIGger:{A B}:PULse:RUNT:THReshold?	Returns the upper and lower thresholds for the A pulse runt trigger
TRIGger:{A B}:PULse:RUNT:WHEN	Sets or queries the type of pulse width the trigger checks for when it uncovers a runt
TRIGger:{A B}:PULse:RUNT:WIDTH	Sets or queries the minimum width for A pulse runt trigger
TRIGger:{A B}:PULse:RUNT?	Returns the current A runt pulse trigger parameters
TRIGger:{A B}:PULse:SOURce	Sets or queries the source for the A pulse trigger

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:{A B}:PULse:TIMEOut:POLarity	Sets or queries the polarity for the A pulse timeout trigger
TRIGger:{A B}:PULse:TIMEOut:POLarity:CH<x>	Sets or queries the polarity for the A or B pulse timeout trigger for the channel
TRIGger:{A B}:PULse:TIMEOut:QUALify	Sets or queries the timeout trigger qualifications
TRIGger:{A B}:PULse:TIMEOut:TIME	Sets or queries the pulse timeout trigger time
TRIGger:{A B}:PULse:TIMEOut?	Returns the parameters for the A pulse timeout trigger
TRIGger:{A B}:PULse:TRANSition:DELTAtime	Sets or queries the delta time used in calculating the transition value
TRIGger:{A B}:PULse:TRANSition:POLarity	Sets or queries the polarity for the A pulse transition trigger
TRIGger:{A B}:PULse:TRANSition:POLarity:CH<x>	Sets or queries the polarity for the A or B pulse transition trigger for the channel
TRIGger:{A B}:PULse:TRANSition:QUALify	Sets or queries the transition trigger qualifications
TRIGger:{A B}:PULse:TRANSition:THReshold:BOTh	Sets the upper and lower thresholds for the transition trigger
TRIGger:{A B}:PULse:TRANSition:THReshold:HIGh	Sets or queries the upper transition trigger threshold
TRIGger:{A B}:PULse:TRANSition:THReshold:LOW	Sets or queries the lower transition trigger threshold
TRIGger:{A B}:PULse:TRANSition:THReshold?	Returns the upper and lower threshold limits for the transition time trigger
TRIGger:{A B}:PULse:TRANSition:WHEN	Sets or queries whether to check for a transitioning signal that is faster or slower than the specified delta time for the transition time trigger
TRIGger:{A B}:PULse:TRANSition?	Returns the delta time, polarity, and both upper and lower threshold limits for the transition time trigger
7TRIGger:{A B}:PULse:WIDth:HIGHLimit	Sets or queries the upper limit for the A pulse width trigger
TRIGger:{A B}:PULse:WIDth:LOWLimit	Sets or queries the lower limit for the pulse width trigger
TRIGger:{A B}:PULse:WIDth:POLarity	Sets or queries the polarity for the pulse width trigger
TRIGger:{A B}:PULse:WIDth:POLarity:CH<x>	Sets or queries the polarity for the A or B pulse width trigger for the channel
TRIGger:{A B}:PULse:WIDth:QUALify	Sets or queries the pulse width trigger qualifications

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:{A B}:PULse:WIDth:WHEn	Sets or queries the criteria for width specification of pulse width trigger events
TRIGger:{A B}:PULse:WIDth?	Returns the trigger pulse width parameters
TRIGger:{A B}:PULse:WINDow?	Returns the current window pulse trigger parameters
TRIGger:{A B}:PULse:WINDow:EVENT	Sets or queries the window trigger event
TRIGger:{A B}:PULse:WINDow:POLarity	Sets or queries the pulse trigger window polarity of the selected trigger Source
TRIGger:{A B}:PULse:WINDow:POLarity:CH<x>	Sets or queries the pulse trigger window polarity of the specified channel
TRIGger:{A B}:PULse:WINDow:QUALify	Sets or queries the window trigger qualifications
TRIGger:{A B}:PULse:WINDow:THReshold:BOTh	Sets both window thresholds
TRIGger:{A B}:PULse:WINDow:THReshold:HIGH	Sets or queries high window threshold
TRIGger:{A B}:PULse:WINDow:THReshold:LOW	Sets or queries low window threshold
TRIGger:{A B}:PULse:WINDow:THReshold?	Returns the upper and lower thresholds for the pulse window trigger
TRIGger:{A B}:PULse:WINDow:WIDTh	Sets or queries minimum width of violation
TRIGger:{A B}:READY?	Queries the trigger ready state
TRIGger:{A B}:TYPE	Sets or queries the type of A or B trigger
TRIGger:{A B}:UPPPerthreshold:CH<x>	Sets or queries the CH<x> upper trigger level for TRIGger:LVLSrcpreference
TRIGger:A:BUS:DATa:FORMat	Sets or queries the trigger data format for the bus
TRIGger:A:BUS:DATa:VALue	Sets or queries the trigger data value depending on the format selected for the bus
TRIGger:A:BUS:I2C:ADDRess:FORMat	Sets or queries the address format for the I2C bus
TRIGger:A:BUS:I2C:ADDRess:MODe	Sets or queries the address mode for the I2C bus
TRIGger:A:BUS:I2C:ADDRess:VALue	Sets or queries the binary address string used for the I2C trigger of the bus
TRIGger:A:BUS:I2C:CONDition	Sets or queries the trigger condition for the I2C bus
TRIGger:A:BUS:I2C:DATa:DIRection	Sets or queries the data direction for the I2C bus
TRIGger:A:BUS:SOUrce	Sets or queries the source for a bus trigger



Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:A:BUS:SPI:CONDition	Sets or queries the trigger condition for the SPI bus
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS}; PULSEForm	Sets or queries the selected communication pulse form
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS};THReshold: HIGH	Sets the communication threshold high level
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS};THReshold: LOW	Sets the communication threshold low level
TRIGger:A:COMMunication:BITRate	Sets or queries the communication bit rate
TRIGger:A:COMMunication:CLOCK: POLarity	Sets or queries the polarity of the clock signal
TRIGger:A:COMMunication:CMI: PULSEForm	Selects the CMI pulse form
TRIGger:A:COMMunication:CODE	Sets the communication signal code
TRIGger:A:COMMunication:SOURce	Sets the communication source channel
TRIGger:A:COMMunication:SOURce:TYPE	Sets the type of communication source channel
TRIGger:A:COMMunication:STANDard	Selects the communication standard that identifies the code and bit rate
TRIGger:A:HOLDoff:ACTUal?	Returns the holdoff time actually used by the A trigger
TRIGger:A:HOLDoff:BY	Sets or queries the type of holdoff for the A trigger
TRIGger:A:HOLDoff:TIME	Sets or queries the A trigger holdoff time
TRIGger:A:HOLDoff?	Returns the A trigger holdoff parameters
TRIGger:A:MODE	Sets or queries the A trigger mode
TRIGger:A:VIDeo?	Returns the video parameters for the A trigger
TRIGger:A:VIDeo:CUSTom?	Returns the custom video parameters for the A trigger
TRIGger:A:VIDeo:CUSTom:FORMat	Sets or queries the video scan format when the video standard is set to Custom
TRIGger:A:VIDeo:CUSTom:LINEPeriod	Sets or queries the custom video line period
TRIGger:A:VIDeo:CUSTom:SYNcInterval	Sets or queries the custom video sync interval
TRIGger:A:VIDeo:FIELD	Sets or queries the video field trigger
TRIGger:A:VIDeo:HOLDoff:FIELD	Sets or queries the trigger holdoff in video field units
TRIGger:A:VIDeo:LINE	Sets or queries the trigger delay as a number of video lines

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:A:VIDeo:POLarity	Sets or queries the polarity of the video trigger
TRIGger:A:VIDeo:SCAN	Sets or queries the video scan rate
TRIGger:A:VIDeo:SOUrce	Sets or queries the video trigger source
TRIGger:A:VIDeo:STANdard	Sets or queries the video standard
TRIGger:AUXLevel	Sets or queries the level for the auxiliary trigger
TRIGger:B	Sets the B trigger level to 50% or queries the B trigger parameters
TRIGger:B:BY	Sets or queries B trigger time or event qualifiers
TRIGger:B:EVENTS:COUNT	Sets or queries the number of events that must occur before the B trigger occurs
TRIGger:B:EVENTS?	Returns the current B trigger events parameter
TRIGger:B:RESET:SOUrce	Sets or queries the source for the trigger reset
TRIGger:B:RESET:STATE	Sets or queries the state of the B trigger
TRIGger:B:RESET:THReshold	Sets or queries the threshold for the trigger reset
TRIGger:B:RESET:TIMEOut	Sets or queries the reset time for the timeout trigger reset
TRIGger:B:RESET:TRANSition	Sets or queries the type of transition required for a transition trigger reset
TRIGger:B:RESET:TYPe	Sets or queries the type of A→B sequential trigger reset
TRIGger:B:SCAN:ADVANCEafter	Sets or queries the B-Event Scan advance after value
TRIGger:B:SCAN:ENABle	Sets or queries if the B-Event Scan is on or off
TRIGger:B:SCAN:ENDevent	Sets or queries the end event value of B-Event Scan
TRIGger:B:SCAN:MODE	Sets or queries the sequence advance mode of B-Event Scan
TRIGger:B:SCAN:STARTevent	Sets or queries the start event value of B-Event Scan
TRIGger:B:STATE	Returns the current state of the triggering system
TRIGger:B:TIME	Sets or queries the B trigger delay time.
TRIGger:ENHanced	Sets or queries the state of trigger position enhancement.
TRIGger:LVLSrcpreference	Sets or queries how the trigger source and levels interact.
TRIGger:QUALification:BUS:FORMat	Sets or queries the format of the bit pattern for the bus qualification.

Table 2-38: Trigger Commands (cont.)

Command	Description
TRIGger:QUALification:BUS:SOUrce	Sets or queries the source for the bus qualification.
TRIGger:QUALification:BUS:VALue	Sets or queries the bit pattern for the bus qualification
TRIGger:SENSITivity	Set or queries the state of trigger sensitivity. When on, the instrument calculates a rolling average for trigger position enhancement
TRIGger:STATE?	Returns 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 may replace VOLts with SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-39: Vertical Commands

Command	Description
AUXIn:BANdwidth	Sets or queries the selectable low-pass bandwidth limit filter of the AUX In input
AUXIn:COUPLing	Sets or queries the input attenuator coupling setting for the AUX In input
AUXIn:OFFSet	Sets or queries the vertical offset for the AUX In input
AUXIn:PRObe:AUTOZero	Sets the probe attached to the AUX In input to autozero
AUXIn:PRObe:DEGAUSS	Starts a degauss cycle of the probe attached to the AUX In input
AUXIn:PRObe:DEGAUSS:STATE?	Returns whether the probe attached to the AUX In input has been degaussed
AUXIn:PRObe:FORCEDRange	Sets the probe attached to the AUX In input to the default range of the probe, or the command queries the default range of the probe
AUXIn:PRObe:GAIN?	Returns the gain factor of the probe that is attached to the AUX In input
AUXIn:PRObe:ID:SERnumber?	Returns the probe serial number
AUXIn:PRObe:ID:TYPE?	Returns the probe type

Table 2-39: Vertical Commands (cont.)

Command	Description
AUXIn:PRObe:INPUTMode	Sets or queries one of the probe's four input modes
AUXIn:PRObe:INPUTMode:AOffset	Sets or queries the requested A mode offset control of the probe that is attached to the AUX In input
AUXIn:PRObe:INPUTMode:BOffset	Sets or queries the requested B mode offset control of the probe that is attached to the AUX In input
AUXIn:PRObe:INPUTMode:CMOffset	Sets or queries the requested common mode offset control of the probe that is attached to the AUX In input
AUXIn:PRObe:INPUTMode:DMOffset	Sets or queries the requested differential mode offset control of the probe that is attached to the AUX In input
AUXIn:PRObe:RANge	Sets or queries the set attenuation range of the probe on the AUX In input
AUXIn:PRObe:RESistance?	Returns the resistance factor of the probe that is attached to the AUX In input
AUXIn:PRObe:SET	Sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors
AUXIn:PRObe:SIGnal	Sets or queries aspects of probe accessory user interfaces
AUXIn:PRObe:TIPtype	Sets or queries the type of probe tip being used
AUXIn:PRObe:UNIts?	Returns a string describing the units of measure for the probe attached to the AUX In input
AUXIn:PROBEFunc:EXTAtten	Sets or queries the input-output ratio, of external attenuation or gain, between the signal and the AUX In input
AUXIn:PROBEFunc:EXTDBatten	Sets or queries the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument AUX In input
AUXIn:PROBEFunc:EXTUnits	Sets or queries the unit of measurement for the external attenuator of the AUX In input
CH<x>?	Returns vertical parameters for the specified channel
CH<x>:BANdwidth	Sets or queries the bandwidth of the specified channel

Table 2-39: Vertical Commands (cont.)

Command	Description
CH<x>:BANdwidth:ENHanced	Sets or queries either the “Bandwidth Enhancement (DSP) Enabled” or “Analog Only” option.
CH<x>:BANdwidth:ENHanced:APPLYtoall	Sets or queries whether the enhanced bandwidth filter applies to all active channels
CH<x>:BANdwidth:ENHanced:FORCe	Sets or queries the Force constant sample rate of the Digital Filters (DSP)
CH<x>:BANdwidth:ENHanced:STATE?	Returns the state of the bandwidth enhancement filter
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>:ICAPture:SOUrce	Selects which digital probe signal will acquire for the specified channel when put in iCapture mode
CH<x>:ICAPture:STATE	Selects the state of the digital probe signal will acquire for the specified channel when put in iCapture mode
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 channel vertical position
CH<x>:PRObe?	Returns the gain, resistance, units, and ID of the probe that is attached to the specified channel
CH<x>:PRObe:AUTOZero	Autozeros the probe attached to the specified channel
CH<x>:PRObe:DEGAUSS	Starts a degauss cycle of the probe attached to the specified channel
CH<x>:PRObe:DEGAUSS:STATE?	Returns whether the probe attached to the specified channel has been degaussed
CH<x>:PRObe:FORCEDRange	Sets the attached probe to its default range, or it queries the default range of the probe
CH<x>:PRObe:GAIN?	Returns the gain of the probe that is attached to the specified channel

Table 2-39: Vertical Commands (cont.)

Command	Description
CH<x>:PRObe:ID?	Returns the type and serial number of the probe that is attached to the specified channel
CH<x>:PRObe:ID:SERnumber?	Returns the serial number of the probe that is attached to the specified channel
CH<x>:PRObe:ID:TYPE?	Returns the type of probe that is attached to the specified channel
CH<x>:PRObe:INPUTMode	Sets or queries the input mode of the probe that is attached to the specified channel
CH<x>:PRObe:INPUTMode:AOffset	Sets or queries the requested A mode offset control of the probe that is attached to the specified channel
CH<x>:PRObe:INPUTMode:BOffset	Sets or queries the requested B mode offset control of the probe that is attached to the specified channel
CH<x>:PRObe:INPUTMode:CMOffset	Sets or queries the requested common mode offset control of the probe that is attached to the specified channel
CH<x>:PRObe:INPUTMode:DMOffset	Sets or queries the requested differential mode offset control of the probe that is attached to the specified channel
CH<x>:PRObe:RANge	Sets or queries the attenuation range of the probe on the designated channel
CH<x>:PRObe:RESistance?	Returns the resistance of the probe that is attached to the specified channel
CH<x>:PRObe:SET	Sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors
CH<x>:PRObe:SIGNAL	Sets or queries aspects of probe accessory user interfaces
CH<x>:PRObe:TIPtype	Sets or queries the type of probe tip being used on the specified channel
CH<x>:PRObe:UNIts?	Returns the units of the probe that is attached to the specified channel
CH<x>:PROBECal?	Returns the probe calibration status
CH<x>:PROBECOntrol	Sets or queries the probe range
CH<x>:PROBEFunc:EXTAtten	Sets the attenuation value for the specified channel to the specified scale factor Or queries the user-specified attenuation

Table 2-39: Vertical Commands (cont.)

Command	Description
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 the unit of measurement for the external attenuator of the specified channel Or queries the user-specified unit of measurement for the external attenuator
CH<x>:SCALE	Sets or queries the vertical scale of the specified channel
CH<x>:TERmination	Sets channel input termination
CH<x>:THRESHold	Sets or queries the comparison level for converting the specified channel to digital form
REF<x>:HORizontal:POSition	Sets or queries the horizontal position of the specified reference waveform in percent of the waveform that is displayed to the right of the center vertical graticule
REF<x>:LABel?	Returns a branch query containing the waveform label name and the coordinates at which the label is displayed
REF<x>:LABel:NAME	Sets or queries the label of the designated waveform
REF<x>:LABel:XPOS	Sets or queries the position of the reference waveform label on the X axis
REF<x>:LABel:YPOS	Sets or queries the position of the reference waveform label on the Y axis
REF<x>:VERTical:POSition	Sets or queries the vertical position of the specified reference waveform
REF<x>:VERTical:SCALE	Sets or queries the reference waveform vertical scale in vertical units/div
SElect?	Returns information on which waveforms are on or off and which waveform is selected
SElect:<wfm>	Turns on the specified waveform Or queries whether the specified channel is on or off
SElect:CH<x>	Sets or queries the displayed state of the specified channel waveform
SElect:CONTROl	Sets or queries the waveform controlled by the front panel

## Waveform Transfer Command Group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points to and 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 may 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 may require more bytes to send the same values than it does with binary. This may 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 into the instrument, you must first specify the record length of the destination waveform record. You do this with the `WFMInpre:NR_Pt` command. Next, specify the first data point within the waveform record. For example, when you set `DATA:START` to 1, data points will be stored starting with the first point in the record. The instrument will ignore the value set by `DATA:STOP` when reading in data. It will stop reading data when there is no more data to read or when it has reached the specified record length.

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.

For `CURVE` queries of FastFrame acquisitions, the data for each frame is concatenated together into one long stream. For example, if you acquire 5 frames of 1000 points each, a curve query (with `DATA:START 1` and `DATA:STOP 1000`) will return 5 \* 1000 or 5000 data points. There is no separator between each frame in the data.

If you set `DATA:START` and `DATA:STOP` to a subset of the record length then you get that section of data from each frame. For example, with 5 frames of 1000 points each and `DATA:START 101` and `DATA:STOP 150`, you will get points 101-150 for the first frame followed by points 101-150 of the second frame, etc. In this case the total number of points returned will be 5 \* 50 or 250 points.

**Waveform Data Locations and Memory Allocation.** The `DATA:SOURce` command specifies the waveform source when transferring a waveform from the instrument. Waveforms sent to the instrument are always stored in one of the four reference memory locations. Use the `DATA:DESTination` command to specify a reference memory location.

**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 [WFMImpre?](#) and [WFMOuppre?](#) 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 `WFMOuppre: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 `WFMOuppre`.
6. Transfer waveform data from the instrument using `CURVe`.

#### **Transferring Waveform Data to the Instrument.**

1. Specify waveform reference memory using `DATA:DESTination`.
2. Set `WFMImpre:NR_Pt` to equal the number of data points to be sent.
3. Specify the waveform data format using `WFMImpre:ENCdg`.
4. Specify the number of bytes per data point using `WFMImpre:BYT_Nr`.
5. Specify first data point in the waveform record using `DATA:START`.
6. Transfer waveform preamble information using `WFMImpre`.
7. Transfer the waveform data using `CURVe`.

---

**NOTE.** *FastAcq waveforms cannot be returned to the instrument via the `CURVe` command.*

---

Table 2-40: Waveform Transfer Commands

Command	Description
<a href="#">ALLOcate:WAVEform:REF&lt;x&gt;?</a>	Returns the record length for the specified reference waveform, if active, or zero (0) if not active
<a href="#">CURVe</a>	The command format transfers waveform data to the instrument (reference memory location specified by <a href="#">DATA:DESTination</a> ) The query format transfers waveform data from instrument specified by the <a href="#">DATA:SOUrce</a> command
<a href="#">CURVENext?</a>	Returns unique waveform data from a new acquisition
<a href="#">CURVEStream</a>	Transfers continuously the waveform data from the instrument as it is acquired
<a href="#">DATA</a>	Sets the format and location of the waveform data that is transferred with the <a href="#">CURVe</a> Command Or queries the format and location of the waveform data that is transferred with the <a href="#">CURVe?</a> command
<a href="#">DATA:SYNCSOURces</a>	Sets or queries if the data sync sources are on or off
<a href="#">DATA:DESTination</a>	Sets or queries the reference memory location for storing waveform data sent to the instrument
<a href="#">DATA:ENCdg</a>	Sets or queries the format of outgoing waveform data
<a href="#">DATA:FRAMESTART</a>	Sets or queries the starting data frame for waveform transfer
<a href="#">DATA:FRAMESTOP</a>	Sets or queries the ending data frame for waveform transfer
<a href="#">DATA:SOUrce</a>	Sets or queries the location of waveform data transferred from the instrument
<a href="#">DATA:START</a>	Sets or queries the starting point in waveform transfer
<a href="#">DATA:STOP</a>	Sets or queries the ending data point in waveform transfer
<a href="#">WAVFrm?</a>	Returns <a href="#">WFMOutpre?</a> and <a href="#">CURVe?</a> data for the waveform as specified by the <a href="#">DATA:SOUrce</a> command
<a href="#">WAVFRMStream?</a>	Returns <a href="#">WFMOUTPRE?</a> and <a href="#">CURVESTREAM?</a> data for the waveforms specified by the <a href="#">DATASOURCE</a> command

Table 2-40: Waveform Transfer Commands (cont.)

Command	Description
WFMInpre?	Returns the waveform formatting specification to be applied to the next incoming CURVE command data
WFMInpre:BIT_Nr	Sets or queries the number of bits per binary waveform point
WFMInpre:BN_Fmt	Sets or queries the format of binary data for the incoming waveform
WFMInpre:BYT_Nr	Sets or queries the binary field data width for the first ordered waveform
WFMInpre:BYT_Or	Sets or queries the byte order of waveform points
WFMInpre:ENCdg	Sets or queries the type of encoding for incoming waveform data
WFMInpre:NR_Pt	Sets or queries the number of points in the transmitted waveform record
WFMInpre:PT_Fmt	Sets or queries the point format of incoming waveform data
WFMInpre:PT_Off	Sets or queries the trigger point within the incoming waveform data
WFMInpre:WFId	Allows a waveform extracted from the instrument to be more easily sent back
WFMInpre:XINcr	Sets or queries the horizontal sampling interval between incoming waveform points
WFMInpre:XUNit	Sets or queries the horizontal units of the incoming waveform
WFMInpre:XZEro	Sets or queries the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMInpre:YMUlt	Sets or queries the vertical scale factor, per digitizing level, of the incoming waveform points
WFMInpre:YOFf	Sets or queries the vertical position of the incoming waveform in digitizing levels
WFMInpre:YUNit	Sets or queries the vertical units of the incoming waveform
WFMInpre:YZEro	Sets or queries the offset of the incoming waveform
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command

Table 2-40: Waveform Transfer Commands (cont.)

Command	Description
WFMOupre:BIT_Nr	Sets or queries the number of bits per waveform point that outgoing waveforms contain
WFMOupre:BN_Fmt	Sets or queries the format of binary data for the waveform
WFMOupre:BYT_Nr	Sets or queries the data width for the waveform
WFMOupre:BYT_Or	Sets or queries the byte order of waveform points
WFMOupre:ENCdg	Sets or queries the type of encoding for outgoing waveforms
WFMOupre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query
WFMOupre:PT_Fmt?	Returns the point format for the waveform
WFMOupre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform
WFMOupre:PT_ORder?	Returns whether the source waveform is DPO
WFMOupre:WFId?	Returns a string describing the acquisition parameters for the waveform
WFMOupre:XINcr?	Returns the horizontal sampling interval
WFMOupre:XUNit?	Returns the horizontal units for the waveform
WFMOupre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger
WFMOupre:YMUIt?	Returns the vertical scale factor per digitizing level for the waveform
WFMOupre:YOff?	Returns the vertical offset in digitizing levels for the waveform
WFMOupre:YUNit?	Returns the vertical units for the waveform
WFMOupre:YZEro?	Returns the vertical offset for the waveform
WFMIupre:NR_FR	Sets or queries one frame, indicating the number of frames in the transmitted waveform record
WFMOupre:NR_FR?	Query-only command returns the number of frames for the DATA:SOURce waveform transmitted in response to a CURVe? query

## Zoom Command Group

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

You can specify up to four zoom areas. You can also specify a zoomed waveform as a zoom source, which enables you to zoom in on a zoomed waveform.

**Table 2-41: Zoom Commands**

Command	Description
ZOOM	Resets the zoom transforms to default values for all traces or live traces. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display
ZOOM:GRATICULE:SIZE	Sets or queries the size of the zoom graticule
ZOOM:HORIZONTAL:POSITION	Sets or queries the horizontal position around which the zoom waveform is displayed. It is freely movable within the confines of the acquired waveform (0% to 100%) and measured from left to right of the acquired waveform
ZOOM:HORIZONTAL:SCALE	Sets or queries the zoom horizontal scale factor around which the zoom waveform is displayed
ZOOM:MATH<x>:HORIZONTAL:POSITION	Sets or queries the horizontal position of the specified math waveform
ZOOM:MATH<x>:HORIZONTAL:SCALE	Sets or queries the zoom horizontal scale factor of the specified math waveform
ZOOM:MATH<x>:VERTICAL:POSITION	Sets or queries the vertical position of the specified math waveform
ZOOM:MATH<x>:VERTICAL:SCALE	Sets or queries the vertical scale of the specified math waveform
ZOOM:MODE	Sets or queries the zoom mode
ZOOM:REF<x>:HORIZONTAL:POSITION	Sets or queries the horizontal position of the specified reference waveform
ZOOM:REF<x>:HORIZONTAL:SCALE	Sets or queries the zoom horizontal scale factor of the specified reference waveform
ZOOM:REF<x>:VERTICAL:POSITION	Sets or queries the vertical position of the specified reference waveform
ZOOM:REF<x>:VERTICAL:SCALE	Sets or queries the zoom vertical scale of the specified reference waveform
ZOOM:SCROLL:DIRECTION	Sets or queries the direction of AutoScroll for zoomed waveforms
ZOOM:SCROLL:LOCK	Sets or queries the state of Scroll Lock, on or off

Table 2-41: Zoom Commands (cont.)

Command	Description
ZOOM:SCROLL:SPEED	Sets or queries the speed of AutoScroll for zoomed waveforms
ZOOM:STATE	Specifies or returns a trace as zoomed on or off
ZOOM:VERTical:POSition	Sets or queries the vertical position around which the zoom waveform is displayed. It is freely movable within the confines of the acquired waveform (0% to 100%) and measured from left to right of the acquired waveform
ZOOM:VERTical:SCAle	Sets or queries the zoom vertical scale factor around which the zoom waveform is displayed
ZOOM:ZOOM<x>	Sets or queries zoom factors for the specified zoom
ZOOM:ZOOM<x>:<wfm>:HORizontal:POSition	Sets or queries the horizontal zoom position for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:HORizontal:SCAle	Sets or queries the horizontal zoom scale of the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:VERTical:POSition	Sets or queries the vertical zoom position for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:VERTical:SCAle	Sets or queries the vertical zoom scale for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:SCROLLLock	Sets or queries Scroll Lock for the specified zoom
ZOOM:ZOOM<x>:<wfm>:DISplay	Sets or queries the display of the specified waveform for the specified zoom
ZOOM:ZOOM<x>:DCHAN:DISplay	Sets or queries the display of the digital waveforms for the specified zoom
ZOOM:ZOOM<x>:STATE	Sets or queries the specified zoom on or off, where x is an integer from 1 to 4 representing the desired zoom window





---

# Commands Listed in Alphabetical Order

## ACQUIRE? (Query Only)

This query-only command returns the following current acquisition parameters:

- Stop after
- Acquisition state
- Mode
- Number of envelopes
- Number of averages
- Number of samples
- Sampling mode

**Group** Acquisition

**Syntax** ACQUIRE?

**Related Commands** [ACQUIRE:MODE](#), [ACQUIRE:NUMFRAMESACQUIRED?](#), [ACQUIRE:NUMAVG](#), [ACQUIRE:NUMENV](#), [ACQUIRE:STOPAFTER](#)

**Examples** ACQUIRE? might return the following string for the current acquisition parameters:  
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT.

## ACQUIRE:INTERPEIGHTBIT

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

**Group** Acquisition

**Syntax** ACQUIRE:INTERPEIGHTBIT {AUTO|ON|OFF}  
ACQUIRE:INTERPEIGHTBIT?

**Arguments** AUTO lets the instrument automatically select the interpolation acquisition mode.  
ON turns on the eight bit interpolation mode.

OFF turns off the eight bit interpolation mode.

**Examples** ACQUIRE:INTERPEIGHTBIT ON sets the interpolation sampling mode ON.  
 ACQUIRE:INTERPEIGHTBIT? might return :ACQUIRE:INTERPEIGHTBIT AUTO indicating the interpolation sampling mode is set to AUTO.

## ACQUIRE:MODE

This command sets or queries the acquisition mode of the instrument. This affects all live waveforms. This command is equivalent to selecting Horizontal/Acquisition from the Horiz/Acq menu, and then choosing the desired mode from the Acquisition Mode group box.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration set by the horizontal scale (time per division). The instrument sampling system always samples at the maximum rate and so an acquisition interval may include more than one sample.

The acquisition mode (which you set using this ACQUIRE:MODE command) determines how the final value of the acquisition interval is generated from the many data samples.

**Group** Acquisition

**Syntax** ACQUIRE:MODE  
 {SAMPLE|PEAKdetect|HIRES|AVERAGE|WFMDB|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. In sample mode, all waveform data has 8 bits of precision. You can request 16 bit data with a CURVE query but the lower-order 8 bits of data will be zero. 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:NUMENV** command.

**WFMDB** (Waveform Database) mode acquires and displays a waveform pixmap. A pixmap is the accumulation of one or more acquisitions.

**ENVELOPE** specifies envelope mode, where the resulting waveform shows the **PEAKDETECT** range of data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the envelope waveform is set or queried using the **ACQUIRE:NUMENV** command.

The instrument acquires data after each trigger event using **Sample** mode; it then determines the pix map location of each sample point and accumulates it with stored data from previous acquisitions.

A Pix map is a two dimensional array. The value at each point in the array is a counter that reflects the hit intensity. Infinite and noninfinite persist display modes affect how pix maps are accumulated. **Zoom**, **Math**, **FastAcq**, **FastFrame**, **XY**, **Roll**, and **Interpolated Time (IT) Sampling Mode** are conflicting features to **WFMDB acqMode**. Turning on one of them generally turns the other one off. Selection of some standard masks (for example, eye masks, which require option **MTM**) changes the acquisition mode to **WFMDB**.

**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 starting acquisition with the **ACQUIRE:STATE RUN** command. This value is reset to zero when any acquisition, horizontal, or vertical arguments that affect the waveform are changed. The maximum number of acquisitions that can be counted is  $2^{30}-1$ . The instrument stops counting when this number is reached. This is the same value that displays in the lower right of the screen.

**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. Use the [ACQuire:MODE](#) command to enable the Average mode. Sending this command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu, selecting the Acquisition tab, and choosing Average from the Acquisition Mode group box. Then enter the desired number of waveforms that will be used to make up an averaged waveform in the # of Wfms box.

**Group** Acquisition

**Syntax** ACQuire:NUMAVg <NR1>  
ACQuire:NUMAVg?

**Related Commands** [ACQuire:MODE](#)

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

**Examples** ACQUIRE:NUMAVG10 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:NUMEnv

This command sets or queries the number of waveform acquisitions that make up an envelope waveform. Sending this command is equivalent to setting the # of Wfms in the Acquisition Setup menu when Envelope Acquisition mode is selected.

---

**NOTE.** Sending *INFINITE* as an argument sets 2e9 envelopes. The query response will be 2e9 not *INFINITE*.

---

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQUIRE:NUMENV {<NR1> INFINITE} ACQUIRE:NUMENV?
<b>Arguments</b>	<NR1> is the number of waveform acquisitions to be enveloped.  INFINITE specifies the number of waveform acquisitions to be enveloped. Setting the argument to infinite sets the envelopes to 2e9.
<b>Examples</b>	ACQUIRE:NUMENV10 specifies that an enveloped waveform will show the result of combining 10 separately acquired waveforms.  ACQUIRE:NUMENV? might return :ACQUIRE:NUMENV 75, indicating that 75 acquisitions are acquired for enveloped waveforms.

## ACQUIRE:NUMFRAMESACQUIRED? (Query Only)

This query-only command returns the number of frames acquired when in FastFrame Single Sequence and acquisitions are running. When the FastFrame single sequence stop condition is set to Stop Manually, this number counts past the number of frames which were requested as new frames are acquired, until STOP is pushed. When not in FastFrame Single Sequence, this value is the same number reported by ACQ:NUMACQ.

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQUIRE:NUMFRAMESACQUIRED?
<b>Related Commands</b>	<a href="#">ACQUIRE:NUMACQ?</a>
<b>Examples</b>	ACQUIRE:NUMFRAMESACQUIRED? might return :ACQUIRE:NUMFRAMESACQUIRED 350, indicating that 350 frames have been acquired.

## ACQUIRE:NUMSAMPLES

This command sets or queries the minimum number of acquired samples that make up a waveform database (WfmDB) waveform for single sequence mode and Mask Pass/Fail Completion Test. This is equivalent to setting the Waveform Database Samples in the Acquisition Mode side menu.

**Group** Acquisition

**Syntax** ACQUIRE:NUMSAMPLES <NR1>  
ACQUIRE:NUMSAMPLES?

**Arguments** <NR1> is the minimum number of acquired samples that make up a waveform database (WfmDB) waveform for single sequence mode and Mask Pass/Fail Completion Test. The default value is 16,000 samples. The range is 5,000 to 2,147,400,000 samples.

In real time mode, if the entire trace is displayed, one triggered acquisition should acquire 5000 samples for a record length of 5000. If you want 5001 samples, the instrument will need to make two acquisitions (10,000 samples). If you request fewer samples than the record length, the number of samples will be the same as the record length.

**Examples** ACQUIRE:NUMSAMPLES5000 specifies that 5000 samples will be acquired in WfmDB mode for single sequence mode and Mask Pass/Fail Completion Test.

ACQUIRE:NUMSAMPLES? might return :ACQUIRE:NUMSAMPLES 10000, indicating that there are 10,000 samples to be acquired in WfmDB mode for single sequence and Mask Pass/Fail Completion Test.

## ACQUIRE:SAMPLINGMODE

This command sets or queries the sampling mode.

**Group** Acquisition

**Syntax** ACQUIRE:SAMPLINGMODE {RT|IT|ET}  
ACQUIRE:SAMPLINGMODE?

**Arguments** RT sets the sampling mode to real time only.

IT sets the sampling mode to interpolation allowed (OFF in REPET).

ET sets the sampling mode to equivalent time allowed (ON in REPET).

The number of channels selected and the sampling speed affects the mode that the instrument uses to acquire data. At slow sampling speeds the instrument uses real time sampling. At fast sampling speeds, the instrument uses equivalent time sampling or interpolation. For speeds between these extremes, some instruments create waveform records differently, depending on the number of input channels selected.

**Examples** ACQUIRE:SAMPLINGMODEIT specifies that interpolation is allowed.

ACQUIRE:SAMPLINGMODE? might return :ACQUIRE:SAMPLINGMODE RT, indicating that sampling mode is real time only.

## 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, pressing the front panel RUN/STOP button is equivalent to sending these commands: ACQUIRE:STOPAFTER RUNSTOP and either ACQUIRE:STATE ON or ACQUIRE:STATE RUN.

**Group** Acquisition

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

**Related Commands** [ACQUIRE:STOPAFTER](#)

**Arguments** OFF stops acquisitions.  
ON starts acquisitions.  
RUN starts acquisitions.  
STOP stops acquisitions.

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

## ALIAS

This command sets or queries the state of alias functionality.

**Group** Alias

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



**Related Commands**    [ALias:STATE](#)

**Arguments**    OFF turns Alias expansion off. If a defined alias is sent when `ALias:STATE` is off, a command error (102) will be generated.

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**    ALIASON 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` has been 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 5;:AUTOSET EXECUTE;:SELECT:CH1 ON" defines an alias named "ST1" that sets up the instrument.  
  
ALIAS:DEFINE? "ST1" returns :ALIAS:DEFINE "ST1",#246 :RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON.

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

**Arguments** OFF turns alias expansion off. If a defined alias is sent when `ALIAS:STATE` is OFF, the instrument generates a command error (102).

ON turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

<NR1> = 0 turns off aliases; any other value turns on aliases.

**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;"`.

## ALLOCate:WAVEform:REF<x>? (Query Only)

This query-only command returns the record length for the specified reference waveform, if active, or zero (0) if not active (that is, no slot exists for the reference waveform).

**Group** Waveform Transfer

**Syntax** `ALLOCate:WAVEform:REF<x>?`

**Examples** ALLOCATE:WAVEFORM:REF3? might return 5000 if a 5000 point waveform was active in the reference 3 slot.

## 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. (To autoselect a video waveform, the video trigger must be set to video standard, not custom. Video arguments require video hardware.) This is equivalent to pressing the front panel AUTOSET button. For a detailed description of autoselect functionality, see *Autoset* in the index of the online help for your instrument.

**Group** Miscellaneous

**Syntax** AUTOSet {EXECute|UNDO|VIDeo|VLines|VFields}

**Related Commands** [DISplay:GRAticule](#)

**Arguments** EXECute autoselects the displayed waveform; this is equivalent to pressing the front panel AUTOSET button. If the display is set to a PAL, MV, or IRE graticule, this argument forces the graticule display to full mode (frame, grid, and cross hair).

UNDO returns the instrument to the setting prior to executing an autoselect.

VIDeo autoselects the displayed waveform.

VLines autoselects the displayed waveform.

VFields autoselects the displayed waveform.

**Examples** AUTOSET VFields sets the instrument to trigger on all video fields.

## AUXIn:BANdwidth

This command sets or queries the selectable low-pass bandwidth limit filter of the AUX In input. The bandwidth will change with an appropriate probe attached. This is equivalent to selecting Bandwidth from the Vertical menu.

The query form of this command always returns the maximum bandwidth of the AUX In input.

**Group** Vertical

**Syntax**     AUXIn:BAWdwidth <NR3>  
 AUXIn:BAWdwidth?

**Arguments**   <NR3> is one of the supported bandwidths on the attached probe.

**Examples**     AUXIN:BAWdWIDTH 10.0000E+9 sets the bandwidth of AUX In input to 10 GHz.  
 AUXIN:BAWdWIDTH?, for DPO7054 instruments, might return :AUXIN:BAWdWIDTH 500.0000E+06, indicating that there is no bandwidth limiting on the AUX In input.

## AUXIn:COUPLing

This command sets or queries the input attenuator coupling setting for the AUX In input. The coupling will change with an appropriate probe attached. This command is equivalent to selecting Coupling from the Vertical menu.

**Group**        Vertical

**Syntax**     AUXIn:COUPLing <NR3>  
 AUXIn:COUPLing?

**Arguments**   <NR3> is one of the supported input attenuator couplings on the attached probe.

**Examples**     AUXIN:COUPLING GND sets the AUX In input to ground.  
 AUXIN:COUPLing? might return :AUXIN:COUPLing GND, indicating that AUX In input is set to GND coupling.

## AUXIn:OFFSet

This command sets or queries the vertical offset for the AUX In input. This command is equivalent to selecting Offset from the Vertical menu.

This command offsets the vertical acquisition window (moves the level at the vertical center of the acquisition window) for the input. Visualize offset as scrolling the acquisition window towards the top of a large signal for increased offset values, and scrolling towards the bottom for decreased offset values. The resolution of the vertical window sets the offset increment for this control.

Offset adjusts only the vertical center of the acquisition window to help determine what data is acquired. The instrument always displays the input signal minus the

offset value. The channel reference marker will move to the vertical graticule position given by the negative of the offset value divided by the scale factor, unless that position is off-screen. If the computed coordinate for the reference mark is off-screen, the mark moves to the nearest screen limit and changes from a right-pointing arrow ( → ) to an arrow pointing in the appropriate off-screen direction.

**Group** Vertical

**Syntax** AUXIN:OFFSet <NR3>  
AUXIN:OFFSet?

**Arguments** <NR3> is the offset value for the specified channel.

**Examples** AUXIN:OFFSet 2.0E-3 sets the offset for AUX In input to 2 mV.  
AUXIN:OFFSet? might return :AUXIN:OFFSET 1.0000E-03, indicating that the offset for AUX In input is set to 1 mV.

## AUXIn:PRObe:AUTOZero (No Query Form)

The command sets the probe attached to the AUX In input to autozero.

**Group** Vertical

**Syntax** AUXIn:PRObe:AUTOZero EXECute

**Arguments** EXECute sets the probe to autozero.

**Examples** AUXIN:PROBE:AUTOZERO EXECUTE sets the probe to autozero.

## AUXIn:PRObe:DEGAUSS (No Query Form)

This command starts a degauss cycle of the probe attached to the AUX In input. The degauss cycle will change with an appropriate probe attached.

**Group** Vertical

**Syntax**    `AUXIn:PRObe:DEGAUSS EXECute`

**Arguments**    `EXECute` starts a probe degauss cycle.

**Examples**    `AUXIN:PROBE:DEGAUSS EXECUTE`, degausses the probe attached to the AUX In input.

## **AUXIn:PRObe:DEGAUSS:STATE? (Query Only)**

This command queries whether the probe attached to the AUX In input has been degaussed.

**Group**    Vertical

**Syntax**    `AUXIn:PRObe:DEGAUSS:STATE?`

**Examples**    `AUXIN:PROBE:DEGAUSS:STATE?` might return  
`AUXIN:PROBE:DEGAUSS:STATE PASSED`, indicating that the probe has been degaussed.

## **AUXIn:PRObe:FORCEDRange**

This command sets the probe attached to the AUX In input to the specified range, or the command queries the probe range.

**Group**    Vertical

**Syntax**    `AUXIn:PRObe:FORCEDRange <NR3>`  
`AUXIn:PRObe:FORCEDRange?`

**Arguments**    `<NR3>` specifies the probe range.

**Examples**    If a TCP0030 current probe is attached the AUX IN input, `AUXIN:PROBE:FORCEDRANGE 5.0` sets the probe to its 5 Ampere range.

`AUXIN:PROBE:FORCEDRANGE` might return `:AUXIN:PROBE:FORCEDRANGE 10.0000E+36`, indicating that the probe range is set to 10.0000E+36 dB.



## AUXIn:PRObe:GAIN? (Query Only)

This query-only command returns the gain factor of the probe that is attached to the AUX In input. The “gain” of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

**Group** Vertical

**Syntax** AUXIn:PRObe:GAIN?

**Examples** AUXIN:PROBE:GAIN? might return :AUXIN:PROBE:GAIN 0.1000E+00, indicating that the attached 10x probe delivers 0.1 V to the AUX In BNC for every 1.0 V applied to the probe input.

## AUXIn:PRObe:ID:SERnumber? (Query Only)

This query-only command returns the probe serial number.

**Group** Vertical

**Syntax** AUXIn:PRObe:ID:SERnumber?

**Examples** AUXIN:PROBE:ID:SERNUMBER? might return :AUXIN:PROBE:ID:SERNUMBER “B999999”, indicating that the probe serial number is set to B999999.

## AUXIn:PRObe:ID:TYPE? (Query Only)

This query-only command returns the probe type.

**Group** Vertical

**Syntax** AUXIn:PRObe:ID:TYPE?

**Examples** AUXIN:PROBE:ID:TYPE? might return :AUXIN:PROBE:ID:TYPE “TAP1500”, indicating that the probe type is set to TAP1500.

## AUXIn:PRObe:INPUTMode

This command sets or queries one of the probe's four input modes on the AUX In input.

**Group** Vertical

**Syntax** AUXIn:PRObe:INPUTMode {DEFauLt|DIFFerentiaL|COMMonmode|A|B}  
AUXIn:PRObe:INPUTMode?

**Arguments** DEFauLt sets the probe to the default mode.  
DIFFerentiaL sets the probe to route differential signals to the host.  
COMMonmode sets the probe to route common-mode signals to the host.  
A sets the probe to route single-ended A signals to the host.  
B sets the probe to route single-ended B signals to the host.

**Examples** AUXIN:PROBE:INPUTMODE A sets the probe to route single-ended A signals to the host.  
AUXIN:PROBE:INPUTMODE? might return :AUXIN:PROBE:INPUTMODE DEFAULT, indicating that the probe input mode on the AUXIN input is set to DEFAULT.

## AUXIn:PRObe:INPUTMode:AOFFSet

This command sets or queries the requested A mode offset control of the probe that is attached to the AUX In input.

**Group** Vertical

**Syntax** AUXIn:PRObe:INPUTMode:AOFFSet <NR3>  
AUXIn:PRObe:INPUTMode:AOFFSet?

**Arguments** <NR3> specifies the A mode offset control.

**Examples** AUXIN:PROBE:INPUTMODE:AOFFSET 0.2 sets the A mode offset control to 0.2 V.

AUXIN:PROBE:INPUTMODE:AOFFSET? might return :AUXIN:PROBE:INPUTMODE:AOFFSET 0.0000, indicating that the A mode offset control is set to 0 V.

## AUXIn:PRObe:INPUTMode:BOFFSet

This command sets or queries the requested B mode offset control of the probe that is attached to the AUX In input.

<b>Group</b>	Vertical
<b>Syntax</b>	AUXIn:PRObe:INPUTMode:BOFFSet <NR3> AUXIn:PRObe:INPUTMode:BOFFSet?
<b>Arguments</b>	<NR3> specifies the B mode offset control.
<b>Examples</b>	AUXIN:PROBE:INPUTMODE:BOFFSET 0.2 sets the B mode offset control to 0.2 V.  AUXIN:PROBE:INPUTMODE:BOFFSET? might return :AUXIN:PROBE:INPUTMODE:BOFFSET? "0.0000", indicating that the B mode offset control is set to 0 V.

## AUXIn:PRObe:INPUTMode:CMOFFSet

This command sets or queries the requested common mode offset control of the probe that is attached to the AUX In input.

<b>Group</b>	Vertical
<b>Syntax</b>	AUXIn:PRObe:INPUTMode:CMOFFSet <NR3> AUXIn:PRObe:INPUTMode:CMOFFSet?
<b>Arguments</b>	<NR3> specifies the common mode offset control
<b>Examples</b>	AUXIN:PROBE:INPUTMODE:CMOFFSET 0.2 sets the common mode offset control to 0.2 V.

AUXIN:PROBE:INPUTMODE:CMOFFSET? might return :AUXIN:PROBE:INPUTMODE:CMOFFSET? "0.0000", indicating that the common mode offset control is set to 0 V.

## AUXIn:PRObe:INPUTMode:DMOFFSet

This command sets or queries the requested differential mode offset control of the probe that is attached to the AUX In input.

**Group** Vertical

**Syntax** AUXIn:PRObe:INPUTMode:DMOFFSet <NR3>  
AUXIn:PRObe:INPUTMode:DMOFFSet?

**Arguments** <NR3> specifies the differential mode offset control.

**Examples** AUXIN:PROBE:INPUTMODE:DMOFFSET 0.2 sets the differential mode offset control to 0.2 V.

AUXIN:PROBE:INPUTMODE:DMOFFSET? might return :AUXIN:PROBE:INPUTMODE:DMOFFSET? "0.0000", indicating that the differential mode offset control is set to 0 V.

## AUXIn:PRObe:RANge

This command controls or queries the set attenuation range of the probe on the AUX In input.

**Group** Vertical

**Syntax** AUXIn:PRObe:RANge  
{ATTEN1X|ATTEN1\_25X|ATTEN1\_5X|ATTEN1\_75X|ATTEN2X|ATTEN2\_5X|ATTEN3X|ATTEN3\_5X|ATTEN4X|ATTEN4\_5X|ATTEN5X|ATTEN5\_5X|ATTEN6X|ATTEN6\_5X|ATTEN7X|ATTEN7\_5X|ATTEN8X|ATTEN8\_5X|ATTEN9X|ATTEN9\_5X|ATTEN10X|ATTEN12\_5X|ATTEN15X|ATTEN17\_5X|ATTEN20X|ATTEN25X|ATTEN30X|ATTEN35X|ATTEN40X|ATTEN45X|ATTEN50X|ATTEN55X|ATTEN60X|ATTEN65X|ATTEN70X|ATTEN75X|ATTEN80X|ATTEN85X|ATTEN90X|ATTEN95X|ATTEN100X|ATTEN125X|ATTEN150X|ATTEN175X|ATTEN200X|ATTEN250X|ATTEN300X|ATTEN350X|ATTEN400X|ATTEN450X|ATTEN500X|ATTEN550X|ATTEN600X|ATTEN650X|ATTEN700X|ATTEN750X|ATTEN800X|ATTEN850X|ATTEN900X|ATTEN950X|ATTEN1000X}  
AUXIn:PRObe:RANge?

**Examples** `AUXIN:PROBE:RANGE ATTEN10X` sets the probe attenuation to 10X.  
`AUXIN:PROBE:RANGE?` might return `:AUXIN:PROBE:RANGE ATTEN1_25X`, indicating that the probe attenuation is set to 25X.

## AUXIn:PRObe:RESistance? (Query Only)

This query-only command returns the resistance factor of the probe that is attached to the AUX In input.

**Group** Vertical

**Syntax** `AUXIn:PRObe:RESistance?`

**Examples** `AUXIN:PROBE:RESISTANCE?` might return `:AUXIN:PROBE:RESISTANCE 10.0000E+06`, indicating that the input resistance of the probe attached to the AUXIN input is set to 10 M $\Omega$ .

## AUXIn:PRObe:SET

This command sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors. The available arguments for this command will vary depending on the accessory you attach to the instrument auxiliary trigger input.

For the P7260 probe, you can select between two attenuation factors using either this GPIB command or the push switch on the probe. The probe enables the relevant path and adjusts the settings based on the characteristics of the path in use. The probe signal path selection is not kept in persistent storage. The probe will lose the selection if you reboot the instrument or remove the probe. Also, the instrument does not store the selection in the save/recall setup operation.

**Group** Vertical

**Syntax** `AUXIn:PRObe:SET <QString>`  
`AUXIn:PRObe:SET?`

**Arguments** `<QString>` is a quoted string representing a settable aspect of the attached accessory.

Example strings for some probes are shown as follows:

**ATTENUATION 5X** sets the P7260 probe to  $\pm 0.75$  V dynamic range with 6 GHz bandwidth and 5X attenuation.

**ATTENUATION 25X** sets the P7260 probe to  $\pm 3$  V dynamic range with 6 GHz bandwidth and 25X attenuation.

**VTERMSOURCE AUTO** sets the P7380SMA probe voltage termination source to auto.

**VTERMSOURCE INTERNAL** sets the P7380SMA probe voltage termination source to internal.

**VTERMSOURCE EXTERNAL** sets the P7380SMA probe voltage termination source to external.

**Examples** **AUXIN:PROBE:SET"ATTEN 25X"** sets the P7260 probe to the selection for  $\pm 3$  V dynamic range with 3 GHz bandwidth and 25X attenuation.

**AUXIN:PROBE:SET?** might return **:CH2:PROBE:SET "ATTENUATION 5X"** indicating that the P7260 probe is set to the selection for  $\pm 0.75$  V dynamic range with 6 GHz bandwidth and 5X attenuation.

**AUXIN:PROBE:SET? "ATTEN"** might return **:CH2:PROBE:SET "ATTENUATION 25X"**, indicating that the P7260 probe is set to the selection for  $\pm 3$  V dynamic range with 3 GHz bandwidth and 25X attenuation.

## AUXIn:PRObe:SIGnal

This command sets or queries aspects of probe accessory user interfaces. The available arguments for this command will vary depending on the accessory you attach to the instrument.

**Group** Vertical

**Syntax** **AUXIn:PRObe:SIGnal {PASS|BYPass}**  
**AUXIn:PRObe:SIGnal?**

**Arguments** **PASS** opens a relay passing your signal to the instrument.

**BYPass** closes a relay preventing your signal from reaching the instrument. During probe degauss, the signal should be bypassed.

**Examples** **AUXIN:PROBE:SIGNALPASS** opens a relay passing your signal to the instrument.

**AUXIN:PROBE:SIGNAL?** might return **:AUXIN:PROBE:SIGNAL BYPASS**, indicating that the relay is closed preventing your signal from reaching the instrument.

## AUXIn:PRObe:TIPType

This command sets or queries the type of probe tip being used.

<b>Group</b>	Vertical
<b>Syntax</b>	AUXIn:PRObe:TIPType {HBWstraightflex OTHER} AUXIn:PRObe:TIPType?
<b>Arguments</b>	HBWstraightflex lets the instrument know you are using a high bandwidth straight-flex probe tip.  OTHER lets the instrument know you are not using a high bandwidth straight-flex probe tip.
<b>Examples</b>	AUXIN:PROBE:TIPTYPE HBWstraightflex sets the auxiliary input probe tip type to HBWstraightflex.  AUXIN:PROBE:TIPTYPE? might return :AUXIN:PROBE:TIPTYPE OTHER, indicating the probe tip is not a HBWstraightflex type.

## AUXIn:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the AUX In input.

<b>Group</b>	Vertical
<b>Syntax</b>	AUXIn:PRObe:UNIts?
<b>Examples</b>	AUXIN:PROBE:UNITS? might return :AUXIN:PROBE:UNITS "V", indicating that the units of measure for the attached probe are volts.

## AUXIn:PROBEFunc:EXTatten

This command sets the input-output ratio, of external attenuation or gain, between the signal and the AUX In input.

The query form of this command returns the user-specified attenuation. Note that this command deals with an attenuation factor, not a gain factor, unlike [CH<x>:PRObe?](#) (This command returns a value independent of the external attenuation). For example, if you specify a 20X attenuation factor, the commands

return the following values (assuming that a 1x probe is presently attached, since the external attenuation is used in combination with the probe attenuation):

```
AUXIN:PROBE:EXTA? 20.00E+0
```

```
AUXIN:PROBE? 1.0E+0
```

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Atten.

**Group** Vertical

**Syntax** AUXIn:PROBEFunc:EXTatten <NR3>  
AUXIn:PROBEFunc:EXTatten?

**Related Commands** [AUXIn:PROBEFunc:EXTDBatten](#)

**Arguments** <NR3> is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

**Examples** AUXIN:PROBEFUNC:EXTATTEN 167.00E-3 sets specifies an external attenuation, which is connected between the user's input signal and the input of the probe attached to the AUX In input.

AUXIN:PROBEFUNC:EXTATTEN? might return :AUXIN:PROBEFUNC:EXTATTEN 1.0000E+00, indicating that the attached probe is connected directly to the user's signal.

## AUXIn:PROBEFunc:EXTDBatten

This command sets the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument AUX In input.

The query form of this command returns the user-specified attenuation in decibels: 1X = 0 dB, 10X = 20 dB, 100X = 40 dB, etc.

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Att(dB).

**Group** Vertical

**Syntax** AUXIn:PROBEFunc:EXTDBatten <NR3>  
AUXIn:PROBEFunc:EXTDBatten?



**Related Commands** [AUXIn:PROBEFunc:EXTDBatten](#)

**Arguments** <NR3> is the attenuation value, which is specified in the range from -200.00 dB to 200.00 dB.

**Examples** AUXIN:PROBEFUNC:EXTDBATTEN 2.5 sets the attenuation value on the AUX IN input to 2.5 dB.

AUXIN:PROBEFUNC:EXTDBATTEN? might return  
:AUXIN:PROBEFUNC:EXTDBATTEEN 2.5000E+00, indicating that the attenuation value on the AUX IN input is set to 2.5 dB.

## AUXIn:PROBEFunc:EXTUnits

This command sets the unit of measurement for the external attenuator of the AUX In input. There is also a corresponding query that returns the user-specified unit of measurement for the external attenuator. Unless these units are set to the factory default string value of "None", they become the attenuated units of measurement for the input. It is assumed that the probe connected to the input is of the correct type to receive the output of the user's external transducer or network.

**Group** Vertical

**Syntax** AUXIn:PROBEFunc:EXTUnits <QString>  
AUXIn:PROBEFunc:EXTUnits?

**Related Commands** [AUXIn:PROBe:UNIts?](#)

**Arguments** <QString> can contain a string of up to eight characters to indicate the attenuation unit of measurement for the AUX In input. However, most instrument attenuators only display the first two characters.

**Examples** AUXIN:PROBEFUNC:EXTUNITS "Pa" sets the unit of measurement for the AUX IN external attenuator.

AUXIN:PROBEFUNC:EXTUNITS? might return :AUXIN:PROBEFUNC:EXTUNITS "Pa", indicating that the AUX IN external attenuator units of measurement are Pascals.

## AUXout? (Query Only)

This query-only command returns the auxiliary output setup and is equivalent to selecting External Signals. From the Utilities menu, and then viewing the current settings for the AUX OUT Configuration.

**Group** Miscellaneous

**Syntax** AUXout?

**Related Commands** [AUXout:SOURce](#)

**Examples** AUXOUT? might return :AUXOUT:SOURCE ATRIGGER;EDGE RISING, indicating that the source at the BNC connector is set to the A trigger and the polarity is set to the rising edge of the trigger output signal.

## AUXout:EDGE

This command sets or queries the direction in which the trigger 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 trigger source at the BNC connection. This command is equivalent to selecting AUX OUT Configuration from the Utilities menu and then selecting the desired Configuration setting.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	AUXout:SOURCE {ATRIGGER BTRIGGER MAIN DELAYED EVENT REFOUT} AUXout:SOURCE?
<b>Related Commands</b>	<a href="#">AUXout?</a>
<b>Arguments</b>	<p>ATRIGGER sets the source at the BNC connector to the main trigger.</p> <p>BTRIGGER sets the source at the BNC connector to the delayed trigger.</p> <p>MAIN sets the source at the BNC connector to the main trigger.</p> <p>DELAYED sets the source at the BNC connector to the delayed trigger.</p> <p>EVENT sets the source at the BNC connector to a specified event.</p> <p>REFOUT sets the source at the BNC connector to the reference output.</p>
<b>Examples</b>	AUXOUT:SOURCE? might return :AUXOUT:SOURCE ATRIGGER, indicating that the source at the BNC connector is set to the A trigger.

## BELI (No Query Form)

This command was previously used to beep an audio indicator and is provided for backward compatibility.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	BEL1
<b>Examples</b>	BELL is accepted but does nothing.

## BUS:B<x>:DISplay:DECOde:FILE

This command sets or queries the name of a TSF-formatted text file to construct a symbolic decode table from for the specified bus.

The decode table associates symbolic names with the numeric values they represent. When a Bus is set to perform symbolic decode, numeric values in its "busform" trace are replaced by the first matching symbol (if any) from the table. Each Bus may use its own specific lookup table. For example, an I2C-type Bus might use a table that associates the symbol "ROM" with the number 0b10100111,

even though another Busses' table associates the symbol "MAX" with that same number. The lookup table format is the same for all the bus types.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

*The decode files for all the buses are available in the default location C:\TekScope\busDecodeTables\.*

---

**Group** Bus

**Syntax** BUS:B<x>:DISPlay:DECOde:FILE {decodeFileName}  
 BUS:B<x>:DISPlay:DECOde:FILE?

**Arguments** decodeFileName specifies the name of a TSF-formatted text file.

**Examples** BUS:B1:DISPLAY:DECODE:FILE  
 "C:\TekScope\busDecodeTables\BusDecode.tsf" uses the TSF-formatted text file BusDecode.tsf.

BUS:B1:DISPLAY:DECODE:FILE? might  
 return :BUS:B1:DISPLAY:DECODE:FILE  
 "C:\TekScope\busDecodeTables\BusDecode.tsf", indicating that the TSF-formatted text file used is BusDecode.tsf.

## BUS:B<x>:DISPlay:DECOde:STAtE

This command sets or queries the enabled/disabled state of the symbolic decode feature for the specified Bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** BUS:B<x>:DISPlay:DECOde:STAtE {OFF|ON|REload}  
 BUS:B<x>:DISPlay:DECOde:STAtE?

**Related Commands** [BUS:B<x>:DISPlay:TYPE](#)

**Arguments** OFF disables the symbolic decode feature.  
 ON enables the symbolic decode feature.  
 REload reparses to whatever symbolic decode filename it is currently assigned. For example, this can be useful if the contents of that file are changed after having assigned it to a Bus.

**Examples** BUS:B1:DISPLAY:DECODE:STATE ON enables the symbolic decode feature.  
 BUS:B1:DISPLAY:DECODE:STATE? might return  
 BUS:B1:DISPLAY:DECODE:STATE? OFF, indicating that the symbolic decode feature is disabled.

## BUS:B<x>:DISplay:FORMat

This command sets or queries the format for displaying numeric values in the 'bus' waveform of the specified bus.

**Group** Bus

**Syntax** BUS:B<x>:DISplay:FORMat {BINary|DECimal|HEXadecimal}  
 BUS:B<x>:DISplay:FORMat?

**Arguments** BINary sets the display format to binary.  
 DECimal sets the display format to decimal.  
 HEXadecimal sets the display format to hexadecimal.

**Examples** BUS:B1:DISPLAY:FORMAT DEC sets the display format for the numerical information in the bus waveform to decimal.  
 BUS:B1:DISPLAY:FORMAT? might return BUS:B1:DISPLAY:FORMAT BINARY, indicating that the display format for the numerical information in the bus waveform is set to binary.

## BUS:B<x>:DISplay:TYPe

This command sets or queries the decoding display style for the specified bus. You can display in either logic waveform or busform or both.

---

**NOTE.** *This command is available only on MSO70000 Series products.*

---

<b>Group</b>	Bus
<b>Syntax</b>	BUS:B<x>:DISPlay:TYPE {WAVEFORMS BUS BOTH} BUS:B<x>:DISPlay:TYPE?
<b>Related Commands</b>	<a href="#">BUS:B&lt;x&gt;:DISPlay:DECOde:STAtE</a>
<b>Arguments</b>	WAVEFORMS, displays the decoding in logic waveforms only. BUS, displays the decoding in busform only. BOTH, displays the decoding in both logic waveform or busform.
<b>Examples</b>	BUS:B1<1>:DISPLAY:TYPE BUS sets the decoding display in busform. BUS:B1<1>:DISPLAY:TYPE? might return :BUS:B1<1>:DISPLAY:TYPE WAVEFORMS, indicating that the decoding display is set to waveform.

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

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

---

**NOTE.** This command is available only on the MSO70000 series of products.

---

<b>Group</b>	Bus
<b>Syntax</b>	BUS:B<x>:I2C:CLOCK:SOURce {CH1 CH2 CH3 CH4 MATH1 MATH2 MATH3 MATH4 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15} [, {CH1 CH2 CH3 CH4 MATH1 MATH2 MATH3 MATH4 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15}] ... BUS:B<x>:I2C:CLOCK:SOURce?
<b>Related Commands</b>	<a href="#">BUS:B&lt;x&gt;:I2C:DATA:SOURce</a> , <a href="#">BUS:B&lt;x&gt;:I2C:RWINADDR</a>
<b>Arguments</b>	CH1–CH4 specifies an analog channel to use as the I2C SCLK source. MATH1–MATH4 specifies a math channel to use as the I2C SCLK source. D0–D15 specifies a digital channel to use as the I2C SCLK source.

**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 D4`, indicating that the I2C SCLK source is set to D4.

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

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

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group**    Bus

**Syntax**    `BUS:B<x>:I2C:DATA:SOURce`  
`{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
`D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} [ ,`  
`{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
`|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}] ...`  
`BUS:B<x>:I2C:DATA:SOURce?`

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

**Arguments**    CH1–CH4 specifies an analog channel to use as the I2C SDATA source.  
MATH1–MATH4 specifies a math channel to use as the I2C SDATA source.  
D0–D15 specifies a digital channel to use as the I2C SDATA source.

**Examples**    `BUS:B1:I2C:DATA:SOURCE D5` sets the I2C SDATA source to D5.  
`BUS:B1:I2C:DATA:SOURCE?` might return `:BUS:B1:I2C:DATA:SOURCE MATH1`, indicating that the I2C SDATA source is set to MATH1.

## BUS:B<x>:I2C:RWINADDR

This command sets or queries the manner in which seven-bit I2C slave addresses are represented in the "busform" display of the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group**    Bus

**Syntax**     `BUS:B<x>:I2C:RWINADDR {No|Yes}`  
`BUS:B<x>:I2C:RWINADDR?`

**Related Commands**     `BUS:B<x>:I2C:CLOCK:SOUrce`, `BUS:B<x>:I2C:DATA:SOUrce`

**Arguments**     `No` 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.

`Yes` displayed 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..0. For example, the slave address byte of 0b10100101 is displayed as the value 0xA5 R.

**Examples**     `BUS:B1:I2C:RWINADDR NO` displays seven-bit slave addresses as integers in the range of 0 to 127.

`BUS:B1:I2C:RWINADDR?` might return `:BUS:B1:I2C:RWINADDR YES`, 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..0.

## BUS:B<x>:LABel

This command sets or queries the waveform label for the specified bus. The bus name string accepts only eight characters and truncates when more than eight characters.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group**     Bus

**Syntax**     `BUS:B<x>:LABel <busName>`  
`BUS:B<x>:LABel?`

**Related Commands**     `BUS:B<x>:POSition`, `BUS:B<x>:TYPE`

**Arguments**     `<busName>` is an alphanumeric string of text, enclosed in quotes. The text string is limited to 30 characters. It contains the text label information for bus.



**Examples** `BUS:B1:LABEL "TEST"` sets the waveform label for the bus B1 to TEST.

`BUS:B1:LABEL?` might return `:BUS:B1:LABEL "BUS 1"`, indicating that the waveform label for the bus B1 is set to "BUS 1".

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

This command sets or queries which edge of a clocked parallel busses' signal establishes the times at which new bus values are sampled.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** `BUS:B<x>:PARAllel:CLOCK:EDGE {RISe|FALL|EITHer}`  
`BUS:B<x>:PARAllel:CLOCK:EDGE?`

**Related Commands** [BUS:B<x>:PARAllel:CLOCK:SOUrce](#), [BUS:B<x>:PARAllel:ISCLOCKED](#), [BUS:B<x>:PARAllel:SOURCES](#)

**Arguments** `RISe` specifies the decoding to happen on the rising edge of the clocked parallel busses' signal.

`FALL` specifies the decoding to happen on the falling edge of the clocked parallel busses' signal.

`EITHer` specifies the decoding to happen on the rising or falling edge of the clocked parallel busses' signal.

**Examples** `BUS:B1:PARALLEL:CLOCK:EDGE FALL` 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 RISE`, 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:SOUrce

This command sets or queries sets or queries the Parallel clock source for the specified bus.

---

**NOTE.** This command is available only on the MSO70000 series of products.

---

<b>Group</b>	Bus
<b>Syntax</b>	<pre>BUS:B&lt;x&gt;:PARAllel:CLOCK:SOURCE {CH1 CH2 CH3 CH4 MATH1 MATH2 MATH3 MATH4 D0 D1 D3  D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15} [, {CH1 CH2 CH3 CH4 MATH1 MATH2 MATH3 MATH4 D0 D1 D2 D3  D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15}] ... BUS:B&lt;x&gt;:PARAllel:CLOCK:SOURCE?</pre>
<b>Related Commands</b>	<a href="#">BUS:B&lt;x&gt;:PARAllel:CLOCK:EDGE</a> , <a href="#">BUS:B&lt;x&gt;:PARAllel:ISCLOCKED</a> , <a href="#">BUS:B&lt;x&gt;:PARAllel:SOURCES</a>
<b>Arguments</b>	<p>CH1–CH4 specifies an analog channel to use as the Parallel clock source.</p> <p>MATH1–MATH4 specifies the math channel to use as the Parallel clock source.</p> <p>D0–D15 specifies a digital channel to use as the Parallel clock source.</p>
<b>Examples</b>	<pre>BUS:B1:PARALLEL:CLOCK:SOURCE D6 sets the Parallel clock source for the bus B1 to D6.  BUS:B1:PARALLEL:CLOCK:SOURCE? might return :BUS:B1:PARALLEL:CLOCK:SOURCE D11, indicating that the Parallel clock source for the bus B1 is set to D11.</pre>

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

This command sets or queries the Parallel bus behavior to either Clocked or not Clocked for the specified bus.

---

**NOTE.** This command is available only on the MSO70000 series of products.

---

<b>Group</b>	Bus
<b>Syntax</b>	<pre>BUS:B&lt;x&gt;:PARAllel:ISCLOCKED {YES NO} BUS:B&lt;x&gt;:PARAllel:ISCLOCKED?</pre>
<b>Related Commands</b>	<a href="#">BUS:B&lt;x&gt;:PARAllel:CLOCK:EDGE</a> , <a href="#">BUS:B&lt;x&gt;:PARAllel:CLOCK:SOURCE</a> , <a href="#">BUS:B&lt;x&gt;:PARAllel:SOURCES</a>

**Arguments** YES specifies the Parallel mode of the specified bus to be clocked.  
NO specifies the Parallel mode of the specified bus to be unlocked (asynchronous).

**Examples** `BUS:B1:PARALLEL:ISCLOCKED YES` sets B1 bus to have Clocked behavior when operating in Parallel mode.  
`BUS:B1:PARALLEL:ISCLOCKED?` might return `:BUS:B1:PARALLEL:ISCLOCKED YES`, indicating that the bus B1 is set to Clocked Parallel mode.

## BUS:B<x>:PARAllel:SOURCES

This command sets or queries the members of the Parallel mode of specified bus according to a supplied list of signals.

The first item on the list becomes the MSB signal of the Bus. The second becomes the next-most-significant, and so on, with the last item becoming the LSB of the bus. If no signals are listed, the Bus is emptied of members.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

*No signal may appear more than once in the list of bus members.*

---

**Group** Bus

**Syntax** `BUS:B<x>:PARAllel:SOURCES`  
{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} [, {CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}] ...  
`BUS:B<x>:PARAllel:SOURCES?`

**Related Commands** [BUS:B<x>:PARAllel:CLOCK:EDGE](#), [BUS:B<x>:PARAllel:CLOCK:SOURce](#)[BUS:B<x>:PARAllel:ISCLOCKED](#)

**Arguments** CH1–CH4 specifies an analog channel to use as the Parallel source.  
MATH1–MATH4 specifies the math channel to use as the Parallel source.  
D0–D15 specifies a digital channel to use as the Parallel source.

**Examples** `BUS:B1:PARALLEL:SOURCES D3, D1, D4, D2` sets the digital signals D3, D1, D4, and D2 as members of the Parallel bus B1.

BUS:B1:PARALLEL:SOURCES? might return :BUS:B1:PARALLEL:SOURCES D3, D5, indicating that the digital signals D3 and D5 are part of the Parallel bus B1.

## BUS:B<x>:POSITION

This command sets or queries the position for the waveform specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** BUS:B<x>:POSITION <NR3>  
BUS:B<x>:POSITION?

**Related Commands** [BUS:B<x>:LABEL](#), [BUS:B<x>:TYPE](#)

**Arguments** <NR3> specifies the position.

**Examples** BUS:B1:POSITION 3 sets the position of the busform of the bus B1 to 3 Divisions. (above the zero level).

BUS:B1:POSITION? might return :BUS:B1:POSITION 5.0000, indicating that the position of the busform of the bus B1 is set to 5 Divisions. (above the zero level).

## BUS:B<x>:SPI:BITOrder

This command sets or queries the shift direction used by an SPI busses' decoding operation to deserialize the data bit sequence into recovered word values.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** BUS:B<x>:SPI:BITOrder {LSB|MSB}  
BUS:B<x>:SPI:BITOrder?

**Related Commands** [BUS:B<x>:SPI:DATA:SIZE](#)

**Arguments** **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 buses 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 B1 to LSB.

**BUS:B1:SPI:BITORDER?** might return **:BUS:B1:SPI:BITORDER MSB**, indicating that the bit order for the SPI bus B1 is set to MSB.

## BUS:B<x>:SPI:CLOCK:POLarity

This command sets or queries the SPI clock (SCK) polarity for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** **BUS:B<x>:SPI:CLOCK:POLarity {RISe|FALL}**  
**BUS:B<x>:SPI:CLOCK:POLarity?**

**Related Commands** [BUS:B<x>:SPI:CLOCK:SOUrce](#)

**Arguments** **RISe** sets the clock to rising edge of the signal.

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

**Examples** **BUS:B1:SPI:CLOCK:POLARITY FALL** sets the SPI clock polarity to falling edge for the bus B1.

**BUS:B1:SPI:CLOCK:POLARITY?** might return **:BUS:B1:SPI:CLOCK:POLARITY RISe**, indicating that the SPI clock polarity for the bus B1 is set to rising edge.

## BUS:B<x>:SPI:CLOCK:SOUrce

This command sets or queries the SPI clock (SCK) source for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** `BUS:B<x>:SPI:CLOCK:SOURce`  
`{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
`D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} [ ,`  
`{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
`|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}] . . .`  
`BUS:B<x>:SPI:CLOCK:SOURce?`

**Related Commands** [BUS:B<x>:SPI:CLOCK:POLarity](#)

**Arguments** CH1–CH4 designates an analog channel as the busses’ SPI clock source.  
 MATH1–MATH4 designates a math waveform as the clock source.  
 D0–D15 designates a digital input signal as the clock source.

**Examples** `BUS:B1:SPI:CLOCK:SOURCE D8` sets the SPI clock source for the bus B1 to D8.  
`BUS:B1:SPI:CLOCK:SOURCE?` might return `:BUS:B1:SPI:CLOCK:SOURCE D11`, indicating that the SPI clock source for the bus B1 is set to D11.

## BUS:B<x>:SPI:DATA:DELay

This command sets or queries the SPI data (DATA) delay for the bus specified by x. The value of x can range from 1 through 16.

It controls the number of SPI data bits the decoding process will ignore at the start of the transfer frame.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** `BUS:B<x>:SPI:DATA:DELay <NR1>`  
`BUS:B<x>:SPI:DATA:DELay?`

**Related Commands** [BUS:B<x>:SPI:DATA:SIZE](#)

**Arguments** <NR1> specifies the SPI bus data delay in bits.

**Examples** `BUS:B1:SPI:DATA:DELAY 5` sets the SPI data delay for the bus B1 to 5 bits.

`BUS:B1:SPI:DATA:DELAY?` might return `:BUS:B1:SPI:DATA:DELAY 3.0000`, indicating that the SPI data delay for the bus B1 is set to 5 bits.

## BUS:B<x>:SPI:DATA:POLARity

This command sets or queries the SPI data (DATA) polarity for the bus specified by x. The value of x can range from 1 through 16. The SPI decode operation treats high inputs as ones in normal polarity and zeros in inverted polarity.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** `BUS:B<x>:SPI:DATA:POLARity {NORMAL|INVERTed}`  
`BUS:B<x>:SPI:DATA:POLARity?`

**Related Commands** [BUS:B<x>:SPI:DATA:DELay](#)

**Arguments** `NORMAL` sets the SPI data polarity to normal.

`INVERTed` sets the SPI data polarity to inverted.

**Examples** `BUS:B1:SPI:DATA:POLARITY NORMAL` sets the SPI data polarity for the bus B1 to normal.

`BUS:B1:SPI:DATA:POLARITY?` might return `:BUS:B1:SPI:DATA:POLARITY INVERTED`, indicating that the SPI data polarity for the bus B1 is set to inverted.

## BUS:B<x>:SPI:DATA:SIZE

This command sets or queries the number of bits per word for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax**     `BUS:B<x>:SPI:DATA:SIZE <NR1>`  
                  `BUS:B<x>:SPI:DATA:SIZE?`

**Related Commands**     [BUS:B<x>:SPI:DATA:DElay](#)

**Arguments**     <NR1> is the data size for the specified bus.

**Examples**     `BUS:B1:SPI:DATA:SIZE 7` sets the data size for the bus B1 to seven bits per word.

`BUS:B1:SPI:DATA:SIZE?` might return `BUS:B1:SPI:DATA:SIZE 8`, indicating that the data size for the bus B1 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 specified by x. The value of x can range from 1 through 16.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group**     Bus

**Syntax**     `BUS:B<x>:SPI:DATA:SOURce`  
                  `{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
                  `D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} [,`  
                  `{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3`  
                  `|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}] ...`  
                  `BUS:B<x>:SPI:DATA:SOURce?`

**Arguments**     CH1–CH4 designates an analog channel as the busses’ SPI data source.  
                  MATH1–MATH4 designates a math waveform as the data source.  
                  D0–D15 designates a digital input signal as the data source.

**Examples**     `BUS:B1:SPI:DATA:SOURCE D2` sets the SPI data source for the bus B1 to D2.

`BUS:B1:SPI:DATA:SOURCE?` might return `:BUS:B1:SPI:DATA:SOURCE D3`, indicating that the SPI data source for the bus B1 is set to D3.



## BUS:B<x>:SPI:SElect:POLarity

This command sets or queries the SPI Slave Select (SS) polarity for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**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** 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 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 high.

## BUS:B<x>:SPI:SElect:SOUrce

This command sets or queries the SPI Slave Select (SS) source for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Bus

**Syntax** BUS:B<x>:SPI:SElect:SOUrce  
 {CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3  
 D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15} [,  
 {CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|D0|D1|D2|D3  
 |D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}] ...  
 BUS:B<x>:SPI:SElect:SOUrce?

**Related Commands**    [BUS:B<x>:SPI:SElect:POLarity](#)

**Arguments**    CH1–CH4 designates an analog channel as the busses’ SPI Slave Select source.  
 MATH1–MATH4 designates a math waveform as the Slave Select source.  
 D0–D15 designates a digital input signal as the Slave Select source.

**Examples**    `BUS:B1:SPI:SELECT:SOURCE MATH5` sets the SPI Slave Select source for the bus B1 to MATH5.  
`BUS:B1:SPI:SELECT:SOURCE?` might return `:BUS:B1:SPI:SELECT:SOURCE D3`, indicating that the SPI Slave Select source for the bus B1 is set to D3.

## BUS:B<x>:TYPE

This command sets or queries the type for the specified bus.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group**    Bus

**Syntax**    `BUS:B<x>:TYPE {PARAllel|I2C|SPI}`  
`BUS:B<x>:TYPE?`

**Arguments**    `PARAllel` specifies the Parallel bus.  
`I2C` specifies the Inter-IC 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.

## 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 a command Commands that Generate an OPC Message(See Table 2-45.).

**Examples** BUSY? might return :BUSY 1, indicating that the instrument is currently busy.

## \*CAL? (Query Only)

This query-only command starts signal path compensation and returns the status upon completion.

**Group** Calibration

**Syntax** \*CAL?

**Related Commands** [CALibrate:RESults:SPC?](#)

**Returns** 0 indicates SPC passed.  
 -1 indicates SPC failed.  
 3 indicates the instrument is still in the warm up phase and SPC was not run.

**Examples** \*CAL? starts the internal signal path calibration and returns the status upon completion.

## CALibrate? (Query Only)

This query returns the status of instrument calibration, including internal signal path calibration, factory calibration, and probe calibration.

<b>Group</b>	Calibration
<b>Syntax</b>	CALibrate?
<b>Examples</b>	CALIBRATE? might return :CALIBRATE:INTERNAL:STATUS PASS;CALIBRATE:FACTORY:STATUS PASS0;:CALIBRATE:RESULTS:SPC PASS;:CALIBRATE:PROBESTATE:CH1 1;CH2 1;CH3 1;CH4 1, indicating the calibration status.

## CALibrate:CALProbe:CH<x>? (Query Only)

This query-only command instructs the instrument to perform a probe calibration for the selected channel and returns the calibration status. The Channel <x> range is 1 through 4. This command is equivalent to selecting Probe Cal from the Vertical menu. You must warm up the instrument for at least 20 minutes before running this command.

---

**NOTE.** *A probe query takes approximately 30 seconds to complete. No other commands will be executed until calibration is complete.*

---

<b>Group</b>	Calibration
<b>Syntax</b>	CALibrate:CALProbe:CH<x>?
<b>Related Commands</b>	CALibrate:PROBestate:CH<x>?, CH<x>:PROBECal?
<b>Returns</b>	<p>H -1 indicating the probe calibration failed for the selected channel.</p> <p>H 0 indicating the probe calibration passed for the selected channel.</p> <p>H 1 indicating the probe calibration is initialized for the selected channel.</p>
<b>Examples</b>	CALibrate:CALProbe:CH2? might return :CALIBRATE:CALPROBE:CH2 0, indicating the probe calibration passed for Channel 2.

## CALibrate:INTERNAL (No Query Form)

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. You can use the [CALibrate:INTERNAL:STATUS?](#) query to return the current status of the internal signal path calibration of the instrument.

**Group** Calibration

**Syntax** CALibrate:INTERNa1

**Related Commands** [CALibrate:RESults:SPC?](#)

**Examples** CALIBRATE:INTERNAL starts the internal signal path calibration of the instrument.

## CALibrate:INTERNAL:START (No Query Form)

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. This command is the same as the [CALibrate:INTERNa1](#) command. You can use the [CALibrate:INTERNAL:STATus?](#) query to return the current status of the internal signal path calibration of the instrument.

**Group** Calibration

**Syntax** CALibrate:INTERNAL:START

**Related Commands** [CALibrate:RESults:SPC?](#)

**Examples** CALIBRATE:INTERNAL:START starts the internal signal path calibration of the instrument.

## CALibrate:INTERNAL:STATus? (Query Only)

This query-only command returns the current status of the internal signal path calibration.

**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 internal signal path calibration run.
  - WARMUP indicates the instrument has not warmed up for the required time to perform signal path calibration.
  - PASS indicates that the signal path calibration completed successfully.
  - FAIL indicates that the signal path calibration did not complete successfully.
  - TEMPDRIFT indicates that the instrument has not reached a state of stable temperature adequate for performing signal path calibration.
  - RUNNING indicates that the signal path calibration is currently running.

**Examples** CALIBRATE:INTERNAL:STATUS? might return  
:CALIBRATE:INTERNAL:STATUS INIT, indicating that the current status of the internal signal path calibration is that it has not been run.

## CALibrate:PRObestate:CH<x>? (Query Only)

This query-only command returns the probe calibration status for the probe of the selected channel, 1 through 4.

**Group** Calibration

**Syntax** CALibrate:PRObestate:CH<x>?

**Related Commands** [CH<x>:PROBECal?](#)

- Returns** This query will return one of the following:
- -1 Indicating the probe calibration failed for the selected channel.
  - 0 Indicating the probe calibration passed for the selected channel.
  - 1 Indicating the probe calibration is initialized for the selected channel.
  - 2 Indicating that the probe calibration for the selected channel is running.

**Examples** :CALIBRATE:PROBESTATE:CH1? might return  
:CALIBRATE:PROBESTATE:CH1 0, indicating that the probe calibration for Channel 1 passed.

## CALibrate:RESuLts? (Query Only)

This query-only command returns the status of internal and factory calibrations, without performing any calibration operations. The results returned do not include the calibration status of attached probes. The `CALibrate:RESuLts?` query is intended to support GO/NoGO testing of the instrument calibration readiness: all returned results should indicate PASS status if the instrument is “fit for duty”. It is quite common, however, to use uncalibrated probes (particularly when the instrument inputs are connected into a test system with coaxial cables).

This command is equivalent to selecting Instrument Calibration from the Utilities menu and then viewing the contents of the Status field.

**Group** Calibration

**Syntax** `CALibrate:RESuLts?`

**Related Commands** [\\*CAL?](#)

**Examples** `CALIBRATE:RESULTS?` returns the status of all the calibration subsystems. The query might return `:CALIBRATE:RESULTS:SPC PASS;FACTORY PASS`, indicating that the status of internal and factory calibrations, without performing any calibration operations is set to PASS.

## CALibrate:RESuLts:SPC? (Query Only)

This query-only command returns the results of the last SPC operation. However, this query does not cause an SPC to be run. This command is equivalent to selecting Instrument Calibration from the Utilities menu.

**Group** Calibration

**Syntax** `CALibrate:RESuLts:SPC?`

**Related Commands** [\\*CAL?](#)

**Examples** `CALIBRATE:RESULTS:SPC?` returns the results of the last SPC operation: either PASS, FAIL, WARMUP, TEMPDRIFT, INIT, or RUNNING.

## CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x. The value of x can range from 1 through 4. This command is similar to selecting Vertical Setup from the Vertical menu.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>?
<b>Examples</b>	<p>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;PROBCAL INIT;PROBE:GAIN 1.0000E+00;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.</p>

## 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 value of x can range from 1 through 4. This is equivalent to selecting Bandwidth from the Vertical menu.

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.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:BANDwidth {TWEnty ONEfi fty TWOfi fty FIVE FULl <NR3>} CH<x>:BANDwidth?
<b>Arguments</b>	<p>TWEnty sets the upper bandwidth limit to 20 MHz.</p> <p>ONEfi fty sets the upper bandwidth limit to 150 MHz.</p> <p>TWOfi fty sets the upper bandwidth limit to 250 MHz.</p> <p>FIVE sets the upper bandwidth limit to 500 MHz.</p> <p>FULl disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.</p>



<NR3> argument is a double-precision ASCII string. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value set the upper bandwidth.

**Examples** CH1: BANDWIDTH TWENTY sets the bandwidth of Channel 1 to 20 MHz.

CH2: BANDWIDTH?, for DPO7054 instruments, might return :CH2: BANDWIDTH 500.0000E+06, indicating that there is no bandwidth limiting on Channel 2.

For DPO7104 instruments, this query might return :CH2: BANDWIDTH 1.0000E+09 indicating that there is no bandwidth limiting on Channel 2.

## CH<x>:BANDwidth:ENHanced

This command sets or queries the “Bandwidth Enhancement (DSP) Enabled” or “Analog Only” option.

---

**NOTE.** When possible, use the *CH<x>:BANDwidth* command to choose the enhanced bandwidth DSP filter.

---

**Group** Vertical

**Syntax** CH<x>: BANDwidth: ENHanced {AUTO|OFF}  
CH<x>: BANDwidth: ENHanced?

**Arguments** AUTO allows “Bandwidth Enhanced (DSP) Enabled”, filtering as possible.  
OFF allows “Analog Only”, preventing DSP filtering.

**Examples** CH1: BANDWIDTH: ENHANCED AUTO allows bandwidth enhancement filtering as possible.

CH1: BANDWIDTH: ENHANCED? might return :CH1: BANDWIDTH: ENHANCED AUTO, indicating that the bandwidth will be enhanced if possible.

## CH<x>:BANDwidth:ENHanced:APPLYtoall

This command sets or queries specified channel’s bandwidth and bandwidth enhanced selection, “DSP” or “Analog Only” to set the other channels the same. Bandwidth achieved on each channel may further be limited while conditions, such as a lower bandwidth probe, that limited the bandwidth exist. A query will almost certainly return 0, as once the request is processed, ApplyToAll will

return to its default state of OFF allowing the channels to continue operating independently.

**Group** Vertical

**Syntax** CH<x>:BANDwidth:ENHanced:APPLYtoall {ON|OFF|1|0}  
CH<x>:BANDwidth:ENHanced:APPLYtoall?

**Arguments** ON sets other channels to bandwidth and bandwidth enhancement filters or analog only of the specified channel.

OFF is the default setting, each channel operates independently regarding bandwidth and bandwidth enhanced (DSP).

1 sets other channels to bandwidth and bandwidth enhancement filters or analog only of the specified channel.

0 is the default and each channel operates independently regarding bandwidth and bandwidth enhanced (DSP).

**Examples** CH1:BANDWIDTH:ENHANCED:APPLYTOALL ON sets the CH1 Bandwidth selection and Digital Filters (DSP) Enabled or Analog Only selection to the other 3 channels. The bandwidth set is influenced by the BW limitations of the channel. If CH1 is not physically limited and has requested 15.0 GHz (DSP) bandwidth, while CH2 has a P7313 probe attached which is limited to 12.5 GHz (HW). A request via CH1 ApplyToAll might result in 13 GHz (DSP) on CH2. If no other Bandwidth selections were made on CH2, subsequent removal of the probe on CH2 would result in 15 GHz (DSP) on CH2 .

CH:BANDWIDTH:ENHANCED:APPLYTOALL? might return  
:CH:BANDWIDTH:ENHANCED:APPLYTOALL 0, indicating that each channel operates independently regarding bandwidth and bandwidth enhanced (DSP).

## CH<x>:BANDwidth:ENHanced:FORCe

This command sets or queries the Force constant sample rate of the Digital Filters (DSP). The bandwidth Enhanced Filter provides you the ability to “Force” constant sample rate, keeping the system in the sample rate base. This is also known as Bandwidth Enhanced AUTO.

---

**NOTE.** *The bandwidth set is influenced by the requested bandwidth and its channel limitations.*

---

---

**NOTE.** *If Force is ON, a request to lower the sample rate below the DSP supported rate is rejected.*

---

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:BANDwidth:ENHanced:FORCE {OFF ON} CH<x>:BANDwidth:ENHanced:FORCE?
<b>Arguments</b>	<p>OFF is the default setting; DSP filtering occurs when sample rate is supported.</p> <p>ON forces the conditions appropriate for Digital Filtering (DSP) to occur. With this option system changes are made to achieve sample rate that Bandwidth Enhanced Digital Filters (DSP) operate in, a Horizontal Mode of Constant Sample Rate, appropriate DSP BW, and Bandwidth Enhanced Auto on the specified channel would all be selected. Incompatible features would be turned off for example, FastAcq would be inhibited.</p>
<b>Examples</b>	<p>CH1:BANDWIDTH:ENHANCED:FORCE? might return :CH1:BANDWIDTH:ENHANCED:FORCE ON, request to lower the sample rate below the DSP supported rate would be rejected.</p>

## CH<x>:BANDwidth:ENHanced:STATE? (Query Only)

This query-only command returns the state of the bandwidth enhancement filter (DSP).

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:BANDwidth:ENHanced:STATE?
<b>Examples</b>	<p>CH1:BANDWIDTH:ENHANCED:STATE? might return :CH1:BANDWIDTH:ENHANCED:STATE 1, indicating the bandwidth enhancement filter is on.</p>

## CH<x>:COUPLing

This command sets or queries the input attenuator coupling setting for the specified channel. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Coupling from the Vertical menu.

---

**NOTE.** *The available arguments depend on the instrument and the attached accessories.*

---

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:COUPLing {AC DC GND DCREJect} CH<x>:COUPLing?
<b>Arguments</b>	AC sets the specified channel to AC coupling. DC sets the specified channel to DC coupling. GND sets the specified channel to ground. Only a flat, ground-level waveform will be displayed. DCREJect sets DC Reject coupling when probes are attached that have that feature.
<b>Examples</b>	CH2:COUPLING GND sets Channel 2 to ground. 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 deskew time for the specified channel. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Deskew from the Vertical Setup menu.

You can adjust the deskew time to add an independent, channel-based delay time to the delay (set by the horizontal position control and common to all channels) from the common trigger point to first sample taken for each channel. This lets you compensate individual channels for different delays introduced by their individual input hook ups.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:DESKew <NR3> CH<x>:DESKew?
<b>Arguments</b>	<NR3> is the deskew time for this channel, ranging from -25 ns to +25 ns with a resolution of 1 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>:ICAPture:SOUrce

This command sets or queries the digital source to acquire iCapture signals.

---

**NOTE.** *To use this command, iCapture should be turned on.*

---

**Group** Vertical

**Syntax** CH<x>:ICAPture:SOUrce {D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15|CQ0}  
 CH<x>:ICAPture:SOUrce?

**Related Commands** [CH<x>:ICAPture:STATE](#)

**Arguments** D0–D15 or CQ0 specifies the digital channel.

**Examples** CH1:ICAPTURE:SOURCE D8 selects the signal from digital probe input D8 as the signal CH1 acquires in iCapture mode (:CH1:ICAPture:STATE is ON)  
 CH1:ICAPTURE:SOURCE? might return CH1:ICAPTURE:SOURCE D9, indicating that the signal from digital probe input is set to D9.

## CH<x>:ICAPture:STATE

This command sets or queries the state of the iCapture feature for the channel specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:ICAPture:STATE {<NR1>|OFF|ON}  
 CH<x>:ICAPture:STATE?

**Related Commands** [CH<x>:ICAPture:SOUrce](#)

**Arguments** <NR1> = 0 turns off the iCapture feature; any other value turns on the iCapture feature.

OFF turns off the iCapture feature.

ON turns on the iCapture feature.

**Examples** CH1:ICAPTURE:STATE ON turns on the iCapture feature.

CH1:ICAPTURE:STATE? might return CH1:ICAPTURE:STATE 0, indicating that the iCapture feature is turned off.

## 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. The value of x can range from 1 through 4. This command is equivalent to selecting Label from the Vertical menu.

**Group** Vertical

**Syntax** CH<x>:LABel:NAME <QString>  
CH<x>:LABel:NAME?

**Arguments** <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 screen offset at which the label (attached to the displayed waveform of the specified channel) is displayed, relative to the left edge of the screen. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Label from the Vertical menu and either viewing or setting X Pos.

**Group** Vertical

<b>Syntax</b>	CH<x>:LABe1:XPOS <NR1> CH<x>:LABe1:XPOS?
<b>Arguments</b>	<NR1> is the location (control in divisions) where the waveform label for the selected channel is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 through 10.
<b>Examples</b>	CH3:LABEL:XPOS 5 moves the waveform label for the Channel 3 that it begins 5 divisions 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 divisions to the right of the left edge of the screen.

## CH<x>:LABe1:YPOS

This command sets or queries the Y screen offset at which the label (attached to the displayed waveform of the specified channel) is displayed, relative to the waveform handle. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Label from the Vertical menu and either viewing or setting Y Pos.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:LABe1:YPOS <NR1> CH<x>:LABe1:YPOS?
<b>Arguments</b>	<NR1> is the location (in divisions) where the waveform label for the selected channel is displayed, relative to the waveform handle. Arguments should be integers ranging from 10 to -10.
<b>Examples</b>	CH3:LABe1:YPOS -2.5 moves the waveform label for the Channel 3 2.5 divisions below the waveform handle.  CH2:LABe1:YPOS? might return :CH2:LABEL:YPOS 0, indicating that the waveform label for the Channel 2 is currently located at the waveform handle.

## CH<x>:OFFSet

This command sets or queries the vertical offset for the specified channel. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Offset from the Vertical menu.

This command offsets the vertical acquisition window (moves the level at the vertical center of the acquisition window) for the selected channel. Visualize offset as scrolling the acquisition window towards the top of a large signal for increased offset values, and scrolling towards the bottom for decreased offset values. The resolution of the vertical window sets the offset increment for this control.

Offset adjusts only the vertical center of the acquisition window for channel waveforms to help determine what data is acquired. The instrument always displays the input signal minus the offset value. The channel reference marker will move to the vertical graticule position given by the negative of the offset value divided by the scale factor, unless that position is off-screen. If the computed coordinate for the reference mark is off-screen, the mark moves to the nearest screen limit and changes from a right-pointing arrow ( → ) to an arrow pointing in the appropriate off-screen direction.

**Group** Vertical

**Syntax** CH<x>:OFFSet <NR3>  
CH<x>:OFFSet?

**Related Commands** [CH<x>:POSition](#)

**Arguments** <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 of the specified channel. The channel is specified by x. The value of x can range from 1 through 4. The position value is applied to the signal before it is digitized. This command is equivalent to selecting Position/Scale from the Vertical menu and either viewing or setting Position.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether it is a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, minus the present offset setting for that channel, are displayed. For example, if the position for Channel 3 is set to 2.0



and the offset is set to 3.0, then input signals equal to 3.0 units are displayed 2.0 divisions above the center of the screen.

**Group** Vertical

**Syntax** CH<x>:POSition <NR3>  
CH<x>:POSition?

**Related Commands** [CH<x>:OFFSet](#), [D<x>:POSITIONMATH<x>:VERTical:POSition](#),  
[REF<x>:VERTical:POSition](#),

**Arguments** <NR3> is the position value, in divisions from the center graticule, ranging from 8 to -8 divisions.

**Examples** CH2:POSition 1.3E+00 positions the Channel 2 input signal 1.3 divisions above the center graticule.

CH1:POSition? might return :CH1:POSITION -1.3000E+00, indicating that the current position of Channel 1 is 1.3 divisions below the center graticule.

## CH<x>:PRObe? (Query Only)

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe?

**Related Commands** [CH<x>:PROBECal?](#)

**Examples** CH2:PROBE? might return :CH2:PROBE:GAIN 1.0000E-01; RESISTANCE 1.0000E+07; UNITS "V"; ID:TYPE "10X" ; SERNUMBER "N/A" for a 10X probe, indicating that (among other parameters) the attenuation factor for the probe attached to Channel 2 is 100.0 mV (assuming that probe units are set to volts).

## CH<x>:PRObe:AUTOZero (No Query Form)

This command sets the probe attached to the input to autozero. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe:AUTOZero EXECute

**Arguments** EXECute sets the probe attached to the specified channel to autozero.

**Examples** CH1:PROBE:AUTOZERO EXECUTE sets the probe attached to the Channel 1 to autozero.

## CH<x>:PRObe:DEGAUSS (No Query Form)

This command starts a degauss cycle of the probe attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUSS EXECute

**Arguments** EXECute starts the degauss cycle.

**Examples** CH1:PROBE:DEGAUSS EXECUTE causes the probe attached to Channel 1 to degauss.

## CH<x>:PRObe:DEGAUSS:STATE? (Query Only)

This command queries whether the probe attached to the specified channel has been degaussed.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUSS:STATE?

**Examples** CH2:PROBE:DEGAUSS:STATE? might return :CH2:PROBE:DEGAUSS:STATE PASSED, indicating that the probe attached to the Channel 2 has been degaussed.

## CH<x>:PRObe:FORCEDRange

This command sets the attached probe to the specified range, or it queries the range of the probe attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

A partial list of probes supported by this command includes the following probes: TCP0001, TCP0030, TCP0150, TDP0500, TDP1000, and TDP1500.

- CH<x>:PRObe:FORCEDRange response will not appear in CH<x>:PRObe? or \*LRN? query responses for channels that do not have probes that implement the appropriate features.
- This command can be sent to any channel whose 'range-selection' policy is Manual without regard to the kind of probe installed in that channel (an error results if the policy is Auto). This permits accepting previous SET? or \*LRN? query responses even if the attached probe is different from the one when the query was sent.
- The CH<x>:PRObe:FORCEDRange? query can be sent explicitly to any channel even though these commands don't appear in a "SET?" response, but the oscilloscope will reply with the string "N/A" if the probe doesn't implement those control mechanisms.

**Group** Vertical

**Syntax** CH<x>:PRObe:FORCEDRange <dynamiciRangeNR3>  
CH<x>:PRObe:FORCEDRange?

**Arguments** <dynamiciRangeNR3> specifies the probe range.

**Examples** If a TCP0030 current probe is attached to the Channel 1 input, CH1:PROBE:FORCEDRANGE 5.0 sets the attached probe to its 5 Ampere range.  
CH1:PROBE:FORCEDRANGE? might return :CH1:PROBE:FORCEDRANGE 2.5000, indicating that the range of the probe attached to the Channel 1 is set to 2.5 Ampere.

## CH<x>:PRObe:GAIN? (Query Only)

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range

from 1 through 4. The “gain” of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

**Group** Vertical

**Syntax** CH<x>:PRObe:GAIN?

**Related Commands** [CH<x>:SCAlE](#)

**Examples** CH2:PROBE:GAIN? might return :CH2:PROBE:GAIN 0.1000E+00, indicating that the attached 10X probe delivers 0.1 V to the Channel 2 BNC for every 1.0 V applied to the probe input.

## CH<x>:PRObe:ID? (Query Only)

This query-only command returns the type and serial of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe:ID?

**Examples** CH2:PROBE:ID? might return :CH2:PROBE:ID:TYPE "10X";SERNUMBER "N/A", indicating that a passive 10X probe of unknown serial number is attached to Channel 2.

## CH<x>:PRObe:ID:SERnumber? (Query Only)

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

---

**NOTE.** For Level 0 and 1 probes, the serial number will be “N/A”.

---

**Group** Vertical

**Syntax** CH<x>:PRObe:ID:SERnumber?

**Examples** CH1:PROBE:ID:SERNUMBER? might return :CH1:PROBE:ID:SERNUMBER "B010289", indicating that the serial number of the probe attached to Channel 1 is B010289.

## CH<x>:PRObe:ID:TYPE? (Query Only)

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4. Level 2 (or higher) probes supply their exact product nomenclature; for Level 0 or 1 probes, a generic 'type string' of "nnX" is returned.

**Group** Vertical

**Syntax** CH<x>:PRObe:ID:TYPE?

**Examples** CH1:PROBE:ID:TYPE? might return :CH1:PROBE:ID:TYPE "P6203", indicating that P6203-type probe is attached to Channel 1.

## CH<x>:PRObe:INPUTMode

This command sets or queries the input mode of the probe that is attached to the specified channel.

**Group** Vertical

**Syntax** CH<x>:PRObe:INPUTMode {DEFault|DIFFerential|COMMonmode|A|B}  
CH<x>:PRObe:INPUTMode?

**Arguments** DEFault sets to the default.  
DIFFerential sets the probe to route differential signals to the host.  
COMMonmode sets the probe to route common-mode signals to the host.  
A sets the probe to route single-ended A signals to the host.  
B sets the probe to route single-ended B signals to the host.

- Examples** CH1:PROBE:INPUTMODE 5.0 sets the attached probe to Channel 1 to its 5 Ampere range.
- CH1:PROBE:INPUTMODE? might return :CH1:PROBE:INPUTMODE A, indicating that the input mode of the probe that is attached to the Channel 1 is set to DEFAULT

## CH<x>:PRObe:INPUTMode:AOffSet

This command sets or queries the requested A mode offset control of the probe that is attached to the specified channel.

- Group** Vertical
- Syntax** CH<x>:PRObe:INPUTMode:AOffSet <NR3>  
CH<x>:PRObe:INPUTMode:AOffSet?
- Arguments** <NR3> specifies the A mode offset control.
- Examples** CH1:PROBE:INPUTMODE:AOFFSET 5.0 sets the requested A mode offset control of the probe that is attached to the Channel 1 to its 5 Ampere range.
- CH1:PROBE:INPUTMODE:AOFFSET? might return :CH1:PROBE:INPUTMODE:AOFFSET? 0.0000, indicating that the requested A mode offset control of the probe that is attached to the Channel 1 is set to 0.0.

## CH<x>:PRObe:INPUTMode:BOFFSet

This command sets or queries the requested B mode offset control of the probe that is attached to the specified channel.

- Group** Vertical
- Syntax** CH<x>:PRObe:INPUTMode:BOFFSet <NR3>  
CH<x>:PRObe:INPUTMode:BOFFSet?
- Arguments** <NR3> specifies the B mode offset control.
- Examples** CH1:PROBE:INPUTMODE:BOFFSET 5.0 sets the requested B mode offset control of the probe that is attached to the Channel 1 to its 5 Ampere range.

CH1:PROBE:INPUTMODE:BOFFSET? might return  
:CH1:PROBE:INPUTMODE:BOFFSET 0.0000, indicating that the requested A  
mode offset control of the probe that is attached to the Channel 1 is set to 0.0.

## CH<x>:PRObe:INPUTMode:CMOFFSet

This command sets or queries the requested common mode offset control of the probe that is attached to the specified channel.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:PRObe:INPUTMode:CMOFFSet <NR3> CH<x>:PRObe:INPUTMode:CMOFFSet?
<b>Arguments</b>	<NR3> specifies the common mode offset control.
<b>Examples</b>	CH1:PROBE:INPUTMODE:CMOFFSET 5.0 sets the requested common mode offset control of the probe that is attached to the Channel 1 to its 5 Ampere range.  CH1:PROBE:INPUTMODE:CMOFFSET? might return :CH1:PROBE:INPUTMODE:CMOFFSET 0.0000, indicating that the requested common mode offset control of the probe that is attached to the Channel 1 is set to 0.0.

## CH<x>:PRObe:INPUTMode:DMOFFSet

This command sets or queries the requested common mode differential mode offset control of the probe that is attached to the specified channel.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:PRObe:INPUTMode:DMOFFSet <NR3> CH<x>:PRObe:INPUTMode:DMOFFSet?
<b>Arguments</b>	<NR3> specifies the differential mode offset control.
<b>Examples</b>	CH1:PROBE:INPUTMODE:DMOFFSET 5.0 sets the requested differential mode offset control of the probe that is attached to the Channel 1 to its 5 Ampere range.

CH1:PROBE:INPUTMODE:DMOFFSET? might return  
 :CH1:PROBE:INPUTMODE:DMOFFSET 0.0000, indicating that the requested differential mode offset control of the probe that is attached to the Channel 1 is set to 0.0.

## CH<x>:PRObe:RANge

This command controls or queries the attenuation range of the probe on the designated channel.

A partial list of probes supported by this command includes the following probes: P7260, P7313, P7313SMA, P7340A, P7360, P7360A, P7380, P7380A, P7380SMA, and P7520.

- CH<x>:PRObe:RANge response will not appear in CH<x>:PRObe? or \*LRN? query responses for channels that do not have probes that implement the appropriate features.
- This command can be sent to any channel whose 'range-selection' policy is Manual without regard to the kind of probe installed in that channel (an error results if the policy is Auto). This permits accepting previous SET? or \*LRN? query responses even if the attached probe is different from the one when the query was sent.
- The CH<x>:PRObe:RANge? query can be sent explicitly to any channel even though these commands do not appear in a SET? response, but the oscilloscope will reply with the string "N/A" if the probe doesn't implement those control mechanisms.

**Group** Vertical

**Syntax** CH<x>:PRObe:RANge <rangeName>  
 CH<x>:PRObe:RANge  
 {ATTEN1X|ATTEN1\_25X|ATTEN1\_5X|ATTEN1\_75X|ATTEN2X|  
 ATTEN2\_5X|ATTEN3X|ATTEN3\_5X|ATTEN4X|ATTEN4\_5X|  
 ATTEN5X|ATTEN5\_5X|ATTEN6X|ATTEN6\_5X|ATTEN7X|  
 ATTEN7\_5X|ATTEN8X|ATTEN8\_5X|ATTEN9X|ATTEN9\_5X|  
 ATTEN10X|ATTEN12\_5X|ATTEN15X|ATTEN17\_5X|ATTEN20X|  
 ATTEN25X|ATTEN30X|ATTEN35X|ATTEN40X|ATTEN45X|  
 ATTEN50X|ATTEN55X|ATTEN60X|ATTEN65X|ATTEN70X|  
 ATTEN75X|ATTEN80X|ATTEN85X|ATTEN90X|ATTEN95X|  
 ATTEN100X|ATTEN125X|ATTEN150X|ATTEN175X|ATTEN200X|  
 ATTEN250X|ATTEN300X|ATTEN350X|ATTEN400X|ATTEN450X|  
 ATTEN500X|ATTEN550X|ATTEN600X|ATTEN650X|ATTEN700X|  
 ATTEN750X|ATTEN800X|ATTEN850X|ATTEN900X|ATTEN950X|  
 ATTEN1000X}  
 CH<x>:PRObe:RANge?



**Examples** CH1:PROBE:RANGE ATTEN10X sets the attenuation range of the probe attached to Channel 1 to 10X.

CH1:PROBE:RANGE? might return :CH1:PROBE:RANGE ATTEN1\_25X, indicating that the attenuation range of the probe on the designated Channel 1 is set to 25X.

## CH<x>:PRObe:RESistance? (Query Only)

This query-only command returns the resistance factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe:RESistance?

**Examples** CH2:PROBE:RESISTANCE? might return :CH2:PROBE:RESISTANCE 1.0000E+06, indicating that the input resistance of the probe attached to Channel 2 is 1 MΩ.

## CH<x>:PRObe:SET

This command sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors. The available arguments for this command will vary depending on the accessory you attach to the instrument.

For the P7260 probe, you can select between two attenuation factors using either this GPIB command or the push switch on the probe. The probe enables the relevant path and adjusts the set of available vertical settings based on the characteristics of the path in use. The probe signal path selection is not kept in persistent storage. The probe will lose the selection if you reboot the instrument or remove the probe. Also, the instrument does not store the selection in the save/recall setup operation.

**Group** Vertical

**Syntax** CH<x>:PRObe:SET <QString>  
CH<x>:PRObe:SET?

**Arguments** <QString> is a quoted string representing a settable aspect of the attached accessory. Example strings for some probes are shown as follows:

**ATTENUation 5X** sets the P7260 probe to  $\pm 0.75$  V dynamic range with 6 GHz bandwidth and 5X attenuation.

**ATTENUation 25X** sets the P7260 probe to  $\pm 3$  V dynamic range with 6 GHz bandwidth and 25X attenuation.

**VTERMSource AUTO** sets the P7380SMA probe voltage termination source to auto.

**VTERMSource INTERNAL** sets the P7380SMA probe voltage termination source to internal.

**VTERMSource EXTERNAL** sets the P7380SMA probe voltage termination source to external.

**Examples** "ATTEN 25X" sets the P7260 probe to the selection for  $\pm 3$  V dynamic range with 3 GHz bandwidth and 25X attenuation.

**CH1:PRObe:SET?** might return **:CH2:PROBE:SET "ATTENUATION 5X"** indicating that the P7260 probe is set to the selection for  $\pm 0.75$  V dynamic range with 6 GHz bandwidth and 5X attenuation.

**CH2:PRObe:SET? "ATTEN"** might return **:CH2:PROBE:SET "ATTENUATION 25X"**, indicating that the P7260 probe is set to the selection for  $\pm 3$  V dynamic range with 3 GHz bandwidth and 25X attenuation.

## CH<x>:PRObe:SIGnal

This command sets or queries aspects of probe accessory user interfaces. The available arguments for this command will vary depending on the accessory you attach to the instrument.

**Group** Vertical

**Syntax** CH<x>:PRObe:SIGna] {PASS|BYPass}  
CH<x>:PRObe:SIGna]?

**Arguments** PASS opens a relay passing your signal to the instrument.

BYPass closes a relay preventing your signal from reaching the instrument. During probe degauss, the signal should be bypassed.

**Examples** **CH2:PROBE:SIGNAL PASS** opens a relay passing your signal on Channel 2 to the instrument.

CH1:PROBE:SIGNAL? might return :CH1:PROBE:SIGNAL BYPASS, indicating that the relay is closed preventing your Channel 1 signal from reaching the instrument.

## CH<x>:PRObe:TIPType

This command sets or queries the type of probe tip being used on the specified channel.

**Group** Vertical

**Syntax** CH<x>:PRObe:TIPType {HBWstraightflex|OTHER}  
CH<x>:PRObe:TIPType?

**Arguments** HBWstraightflex lets the instrument know you are using a high bandwidth straight-flex probe tip.  
  
OTHER lets the instrument know you are not using a high bandwidth straight-flex probe tip.

**Examples** CH1:PROBE:TIPTYPE HBWSTRAIGHTFLEX sets the channel 1 probe tip type to HBWstraightflex.  
  
CH1:PROBE:TIPTYPE? might return :CH1:PROBE:TIPTYPE OTHER, indicating the probe tip is not an HBWstraightflex type.

## CH<x>:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Vertical

**Syntax** CH<x>:PRObe:UNIts?

**Related Commands** [CH<x>:PROBEFunc:EXTUnits](#)

**Examples** CH4:PROBE:UNITS? might return :CH4:PROBE:UNITS "V", indicating that the units of measure for the probe attached to Channel 4 are volts.

## CH<x>:PROBECal? (Query Only)

This query-only command returns the probe calibration state for the selected channel. The channel is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Probe Cal from the Vertical menu.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:PROBECal?
<b>Returns</b>	This query will return one of the following: <ul style="list-style-type: none"> <li>■ FAIL signifies that the probe calibration has failed for the selected channel.</li> <li>■ INIT signifies that the probe calibration has not yet been run for the selected channel.</li> <li>■ PASS signifies the probe calibration has passed for the selected channel.</li> </ul>
<b>Examples</b>	CH2:PROBECAL? might return :CH2:PROBECAL PASS indicating that the probe calibration has passed for Channel 2.

## CH<x>:PROBECOntrol

This command allows you to set or query the probe range.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:PROBECOntrol {AUTO MANual} CH<x>:PROBECOntrol?
<b>Arguments</b>	AUTO sets the values. The probe range is automatically calculated.  MANual allows you to select various valid values for the probe connected to a particular channel.
<b>Examples</b>	CH2:PROBECONTROL AUTO sets the values and the probe range is automatically calculated.  CH2:PROBECONTROL? might return :CH2:PROBECONTROL MANUAL indicating that you can select various valid values for the probe connected to a particular channel.

## CH<x>:PROBEFunc:EXTAtten

This command is used to specify the input-output ratio, of external attenuation or gain, between the signal and the instrument input channels.

The channel is specified by x. The value of x can range from 1 through 4.

The query form of this command returns the user-specified attenuation. Note that this command deals with an attenuation factor, not a gain factor, unlike [CH<x>:PROBe?](#) (This command returns a value independent of the external attenuation). For example, if you specify a 20x attenuation factor for channel 1, the commands return the following values (assuming that a 1x probe is presently attached, since the external attenuation is used in combination with the probe attenuation):

```
CH1:PROBe:EXTA? 20.00E+0
```

```
CH1:PROBe? 1.0E+0
```

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Atten.

**Group** Vertical

**Syntax** CH<x>:PROBEFunc:EXTAtten <NR3>  
CH<x>:PROBEFunc:EXTAtten?

**Related Commands** [CH<x>:PROBEFunc:EXTDBatten](#)

**Arguments** <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 user's 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

Use this command to specify 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 value of x can range from 1 through 4.

The query form of this command returns the user-specified attenuation in decibels: 1X = 0 dB, 10X = 20 dB, 100X = 40 dB, etc.

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Att (dB).

**Group** Vertical

**Syntax** CH<x>:PROBEFunc:EXTDBatten <NR3>  
CH<x>:PROBEFunc:EXTDBatten?

**Related Commands** [CH<x>:PROBEFunc:EXTAtten](#)

**Arguments** <NR3> is the attenuation value, which is specified in the range from -200.00dB 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 the unit of measurement for the external attenuator of the specified channel. The channel is specified by x. The value of x can range from 1 through 4. There is also a corresponding query that returns the user-specified unit of measurement for the external attenuator. Unless these units are set to the factory default string value of "None", they become the attenuated units of measurement for that channel. It is assumed that the probe connected to the specified channel is of the correct type to receive the output of the user's external transducer or network.

**Group** Vertical

**Syntax** CH<x>:PROBEFunc:EXTUnits <QString>  
CH<x>:PROBEFunc:EXTUnits?

**Related Commands** [CH<x>:PRObe:UNIts?](#)

**Arguments** <QString> can contain a string of up to eight characters to indicate the attenuation unit of measurement for the specified channel. However, most instrument attenuators only display the first two characters.

**Examples** CH4:PROBEFUNC:EXTUNITS "Pa" sets the unit of measurement for the Channel 4 external attenuator.

CH2:PROBEFUNC:EXTUNITS? might return :CH2:PROBEFUNC:EXTUNITS "Pa", indicating that the Channel 2 external attenuator units of measurement are Pascals.

## CH<x>:SCALE

This command sets or queries the vertical scale of the specified channel. The channel is specified by x. The value of x can range from 1 through 4. Sending this command is equivalent to selecting Vertical Setup from the Vertical menu and then viewing or setting the Scale.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects channel waveforms differently from other waveforms:

- For channel waveforms, this setting controls the vertical size of the acquisition window and the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.
- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

**Group** Vertical

**Syntax** CH<x>:SCALE <NR3>  
CH<x>:SCALE?

**Related Commands** [CH<x>:OFFSet](#), [CH<x>:POSition](#), [REF<x>:VERTical:SCALE](#), [MATH<x>:VERTical:SCALE](#), [CH<x>:THRESHold](#)

**Arguments** <NR3> is the vertical channel scale in units per division.

- Examples** CH4:SCALE 100E-03 sets the Channel 4 scale to 100 mV per division.  
 CH2:SCALE? might return :CH2:SCALE 1.0000E+00, indicating that the current scale setting of Channel 2 is 1 V per division.

## CH<x>:TERmination

This command sets the connected/disconnected status of a 50  $\Omega$  resistor, which may be connected between the specified channel's coupled input and instrument ground. The channel is specified by x. The value of x can range from 1 through 4. There is also a corresponding query that requests the termination parameter and translates this enumeration into one of the two float values. This command is equivalent to selecting Termination from the Vertical menu or toggling between termination values from the VERTICAL area of the front panel.

---

**NOTE.** *The available arguments depend on the instrument and the attached accessories.*

---

- Group** Vertical
- Syntax** CH<x>:TERmination <NR3>  
 CH<x>:TERmination?
- Arguments** <NR3> specifies the channel input resistance, which can be specified as 50  $\Omega$  or 1,000,000  $\Omega$ .
- Examples** CH4:TERMINATION 50.0E+0 establishes 50  $\Omega$  impedance on Channel 1.  
 CH2:TERMINATION? might return :CH2:TERMINATION 50.0E+0, indicating that Channel 2 is set to 50  $\Omega$  impedance.

## CH<x>:THRESHold

This command sets or queries the comparable threshold for converting the analog signal to digital form for the channel specified by x. The value of x can range from 1 through 4.

- Group** Vertical
- Syntax** CH<x>:THRESHold <NR3>  
 CH<x>:THRESHold?



<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:POSition</a> , <a href="#">CH&lt;x&gt;:SCAlE</a> <a href="#">D&lt;x&gt;:THRESHold</a> , <a href="#">MATH&lt;x&gt;:THRESHold</a>
<b>Arguments</b>	<NR3> specifies the analog threshold, in volts.
<b>Examples</b>	CH1:THRESHOLD 1 sets the threshold value of the channel CH1 to 1 volt. CH1:THRESHOLD? might return CH1:THRESHOLD 500.0000E-3, indicating that the threshold value set for the channel CH1 is 0.5 volts.

## \*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) GPIB 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.

## CMDBatch

This command sets or queries the state of command batching. By batching commands, database transactions can be optimized, increasing command throughput. Also, batching allows for ALL commands in an individual batch to

be order independent and accomplish the same result as if the commands were coupled.

The Batch state is persistent and will be saved across power cycles, but will not be saved and recalled as part of a setup. In a setup scenario, the factory initial value is enabled.

**Group** Miscellaneous

**Syntax** `CMDBatch {OFF|ON|<NR1>}`  
`CMDBatch?`

**Arguments** OFF turns command batching off.  
 ON turns command batching on.  
 <NR1> = 0 turns command batching off; any other value turns command batching on.

**Examples** `CMDBATCHOFF` disables command batching.  
`CMDBATCH?` might return `:CMDBATCH 1`, indicating that command batching is turned on.

## CQ<x>:THRESHold

This command sets or queries the comparable threshold for converting the clock/qualifier signal to its digital form, where x is zero.

**Group** Digital

**Syntax** `CQ<x>:THRESHold <NR3>`  
`CQ<x>:THRESHold?`

**Arguments** <NR3> specifies the channel/qualifier threshold, in volts.

**Examples** `CQ0:THRESHOLD 0.5` sets the threshold value of the channel/qualifier CQ0 to 0.5 volt.  
`CQ0:THRESHOLD?` might return `CQ0:THRESHOLD 1.4000`, indicating that the threshold value set for the channel/qualifier CQ0 is 1.4 volts.

## CURSor?

This query-only command returns all of the current cursor settings.

**Group** Cursor

**Syntax** CURSor?

**Examples** CURSOR? might return the following as the current cursor settings:

```
:CURSOR:STATE 1;FUNCTION VBARS;LINESTYLE SOLID;MODE
INDEPENDENT;SOURCE1 CH1; SOURCE2 CH1;SCREEN:STYLE
LINES; :CURSOR:VBARS:UNITS SECONDS;POSITION1
-1.6000E-6;POSITION2 1.6000E-6; :CURSOR:HBARS:POSITION1
300.0000E-3; POSITION2 -300.0000E-3;UNITS BASE;
:CURSOR:WAVEFORM:POSITION1 -1.6000E-6;POSITION2 1.6000E-6;
SOURCE2 CH1;UNITS BASE;STYLE LINE_X;:CURSOR:XY :PRODELTA
99.1000E+36;PRODUCT1 99.1000E+36;PRODUCT2 99.1000E+36;
READOUT RECTANGULAR;RADIUS1 99.1000E+36;RADIUS2 99.1000E+36;
RATDELTA 99.1000E+36;RATIO1 99.1000E+36;RATIO2 99.1000E+36;
RDELTA 99.1000E+36;RECTX1 0.0000;RECTX2 0.0000;RECTY1
0.0000; RECTY2 0.0000;THDELTA 99.1000E+36;THETA1
99.1000E+36; THETA2 99.1000E+36;XDELTA 0.0000;YDELTA 0.0000.
```

## CURSor:FUNction

This command sets or queries the cursor type. Sending this command is equivalent to selecting Cursor Type from the Cursors menu, and then choosing from the drop-down list.

**Group** Cursor

**Syntax** CURSor:FUNction {OFF|HBARs|VBARs|SCREEN|WAVEform}  
CURSor:FUNction?

**Related Commands** [CURSor:STATE](#)

**Arguments** OFF removes the cursors from the display but does not change the cursor type.  
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**     `CURSOR:FUNCTIONVBARS` selects the vertical bar cursors type.

`CURSOR:FUNCTION?` might return `:CURSOR:FUNCTION HBARS`, indicating that the horizontal bar cursors are currently selected.

## **CURSOR:HBARS? (Query Only)**

This query-only command returns the current settings for the horizontal bar cursors.

**Group**     Cursor

**Syntax**     `CURSOR:HBARS?`

**Examples**     `CURSOR:HBARS?` might return the horizontal bar setting as `:CURSOR:HBARS:POSITION1 320.0000E-03;POSITION2-320.0000E-03;UNITS BASE`.

## **CURSOR:HBARS:DELTA? (Query Only)**

This query-only command returns the vertical difference between the two horizontal bar cursors.

**Group**     Cursor

**Syntax**     `CURSOR:HBARS:DELTA?`

**Related Commands**     [CURSOR:HBARS:UNITS?](#)

**Returns**     A floating point value with an exponent.

**Examples** `CURSOR:HBARS:DELTA?` might return `:CURSOR:HBARS:DELTA 5.0800E+00`, indicating that the voltage difference between the two cursors is 5.08 V.

## **CURSor:HBARs:POSITION<x>**

This command sets or queries the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). The cursor is specified by *x*, which can be 1 or 2. This command is the equivalent to selecting Cursor Position from the Cursors menu, selecting the H Bars Cursor Type, and then viewing or editing the desired cursor position.

**Group** Cursor

**Syntax** `CURSor:HBARs:POSITION<x> <NR3>`  
`CURSor:HBARs:POSITION<x>?`

**Related Commands** [CURSor:FUNcTion](#)

**Arguments** `<NR3>` specifies the cursor position relative to ground.

**Examples** `CURSOR:HBARS:POSITION1 25.0E-3` positions Cursor 1 of the horizontal cursors at 25 mV.

`CURSOR:HBARS:POSITION2?` might return `:CURSOR:HBARS:POSITION2 -64.0000E-03`, indicating that Cursor 2 of the horizontal bar cursors is at -64 mV.

## **CURSor:HBARs:UNIts? (Query Only)**

This query-only command returns the units for the horizontal bar cursors.

This query always returns `BASE`, indicating that the units for the horizontal bar cursors are those of the base waveform to which they were assigned.

**Group** Cursor

**Syntax** `CURSor:HBARs:UNIts?`

**Examples** `CURSOR:HBARS:UNITS?` always returns `:CURSOR:HBARS:UNITS BASE` indicating that the units for the horizontal bar cursors are base.

## CURSor:LINEStyle

This command sets or queries the cursors line style used when cursors are displayed.

**Group** Cursor

**Syntax** CURSor:LINEStyle {SOLID|DASHed|SDASHed}  
CURSor:LINEStyle?

**Arguments** SOLID displays the cursors as solid lines.  
DASHed displays the cursors as dashed lines.  
SDASHed displays the cursors as solid-dashed lines.

**Examples** CURSOR:LINESTYLESOLID displays the cursors as solid lines.  
CURSOR:LINESTYLE? might return :CURSOR:LINESTYLE DASHED, indicating that the cursors are displayed as dashed lines.

## CURSor:MODE

This command sets or queries whether the two cursors move together in unison or separately. This command is the equivalent of selecting Cursor Mode from the Cursors menu, and then choosing from the drop-down list.

**Group** Cursor

**Syntax** CURSor:MODE {TRACk|INDEpendent}  
CURSor:MODE?

**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** CURSOR:MODETRACk specifies that the cursor positions move in unison.  
CURSOR:MODE? might return :CURSOR:MODE TRACk, indicating that the two cursors move in unison.

## CURSor:SCREEN:STYle

This command sets or queries the cursor type for Screen mode.

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:SCREEN:STYle {LINE_X LINES X} CURSor:SCREEN:STYle?
<b>Arguments</b>	LINE_X specifies the cursor style to be a line with superimposed X. LINES specifies the cursor style to be a line. X specifies the cursor style to be an X.
<b>Examples</b>	CURSOR:SCREEN:STYLELINES sets the cursor style to lines. CURSOR:SCREEN:STYLE? might return :CURSOR:SCREEN:STYLE X, indicating that the style for screen cursors is X.

## CURSor:SCREEN:XPOSITION<x>

This command sets or queries the x position of the specified screen cursor. The cursor is specified by x, which can be 1 or 2. If the cursor is not specified, it defaults to cursor 1.

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:SCREEN:XPOSITION<x> <NR3> CURSor:SCREEN:XPOSITION<x>?
<b>Related Commands</b>	<a href="#">CURSor:SCREEN:YPOSITION&lt;x&gt;</a>
<b>Arguments</b>	<NR3> specifies the x position of the specified screen cursor.
<b>Examples</b>	CURSOR:SCREEN:XPOSITION1 15.0E-3 sets the x position of the screen Cursor 1 to 15 mV. CURSOR:SCREEN:XPOSITION2? might return :CURSOR:SCREEN:XPOSITION2 -64.0000E-03 indicating that x position of the screen Cursor 2 is set to -64 mV.

## CURSor:SCREEN:YPOSITION<x>

This command sets or queries the y position of the specified screen cursor. The cursor is specified by x, which can be 1 or 2. If the cursor is not specified, it defaults to cursor 1.

**Group** Cursor

**Syntax** CURSor:SCREEN:YPOSITION<x> <NR3>  
CURSor:SCREEN:YPOSITION<x>?

**Related Commands** [CURSor:SCREEN:XPOSITION<x>](#)

**Arguments** <NR3> specifies the y position of the specified screen cursor.

**Examples** CURSOR:SCREEN:YPOSITION1 25.0E-3 sets the y position of the screen Cursor 1 to 25 mV.

CURSOR:SCREEN:YPOSITION2? might return :CURSOR:SCREEN:YPOSITION2 -53.0000E-03 indicating that y position of the screen Cursor 2 is set to -53 mV.

## CURSor:SOUrce<x>

This command sets or queries the source(s) for the currently selected cursor type. The cursor is specified by x, which can be 1 or 2. If the cursor is not specified, it defaults to cursor 1.

This command is equivalent to selecting Cursor Setup from the Cursors menu, and then choosing the desired cursor source.

**Group** Cursor

**Syntax** CURSor:SOUrce<x>  
{CH1|CH2|CH3|CH4|MATH1|MATH2|MATH3|MATH4|REF1|REF1|REF3|REF4}  
CURSor:SOUrce<x>?

**Arguments** CH1-CH4 sets an input channel waveform as the cursor source. The valid channel waveform range is from 1 through 4.

MATH1-MATH4 sets a math waveform as the cursor source. The valid math waveform range is from 1 through 4.



REF1–REF4 sets a reference waveform as the cursor source. The valid reference waveform range is from 1 through 4.

**Returns** Any valid waveform.

**Examples** `CURSOR:SOURCE 1 CH2` sets the Cursor1 source to Channel 2.

`CURSOR:SOURCE 2?` might return `:CURSOR:SOURCE2 CH2` indicating that the Cursor2 source is Channel 2.

`CURSOR:SOURCE?` might return `:CURSOR:SOURCE1 CH2`, indicating that the Cursor1 source is Channel 2. If you do not specify the cursor, the cursor 1 source is returned.

## CURSor:STATE

This command sets or queries the state of cursors. Note that setting the cursor state does not modify the cursor type.

This command is equivalent to pressing the CURSOR button on the front panel.

**Group** Cursor

**Syntax** `CURSor:STATE {ON|OFF|<NR1>}`  
`CURSor:STATE?`

**Related Commands** [CURSor:FUNCTion](#)

**Arguments** ON displays the cursors.

OFF removes cursors from the display.

<NR1> = 0 disables the display cursors; any other value enables the display cursors.

**Examples** `CURSOR:STATEON` displays the cursors.

`CURSOR:STATE?` might return `:CURSOR:STATE 1`, indicating that the cursors are active on the display.

## CURSor:VBArS

This command sets or queries the current settings for the vertical bar cursors or forces them to snap to positions specified by the [DATA:START](#) and [DATA:STOP](#) commands.

**Group** Cursor

**Syntax** CURSor:VBArS SNAp  
CURSor:VBArS?

**Related Commands** [DATA:START](#), [DATA:STOP](#)

**Arguments** SNAp forces the position of vertical bar cursor 1 and 2 to snap to the waveform record points specified by the [DATA:START](#) and [DATA:STOP](#) commands, respectively.

**Examples** CURSOR:VBARSSNAp forces vertical bar cursor1 to the DATA:START position and vertical bar cursor 2 to the DATA:STOP position.

CURSOR:VBARS? might return the following vertical bar settings  
:CURSOR:VBARS:UNITS SECONDS;POSITION1 1.0000E-06;POSITION2  
9.0000E-06.

## CURSor:VBArS:DELTA? (Query Only)

This query-only command returns the horizontal difference between the two vertical bar cursors. The units are specified by the [CURSor:VBArS:UNItS](#) command. The position can appear in units of Base or 1/Base.

**Group** Cursor

**Syntax** CURSor:VBArS:DELTA?

**Related Commands** [CURSor:VBArS:UNItS](#)

**Returns** A floating point value with an exponent.

**Examples** CURSOR:VBARS:DELTA? might return :CURSOR:VBARS:DELTA 1.0640E+00, indicating that the time between the vertical bar cursors is 1.064 s.

## CURSor:VBArS:POSItION<x>

This command sets or queries the horizontal position for vertical bar cursors. The cursor is specified by x, which can be 1 or 2. Values are with respect to trigger position or the zero reference point for the designated waveform (if horizontal units are not set to time). Use the [CURSor:VBArS:UNItS](#) command to specify units. The position can appear in units of base or 1/base.

This command is the equivalent of selecting Cursor Setup from the Cursors menu, selecting V Bars Cursor Type, and then viewing or editing the desired cursor position.

**Group** Cursor

**Syntax** CURSor:VBArS:POSItION<x> <NR3>  
CURSor:VBArS:POSItION<x>?

**Related Commands** [CURSor:VBArS:UNItS](#)

**Arguments** <NR3> specifies the cursor position.

**Returns** A floating point value with an exponent.

**Examples** CURSor:VBArS:POSItION2 9.00E-6 positions the cursor2 vertical bar cursor at 9 ms.

CURSor:VBArS:POSItION1? this command might return  
:CURSOR:VBARS:POSITION1 1.0000E-06, indicating that the  
cursor1 vertical bar is positioned at 1 ms.

## CURSor:VBArS:POS<x>

This command sets or queries the horizontal position for vertical bar cursors. The cursor is specified by x, which can be 1 or 2. Values are with respect to trigger position or the zero reference point for the designated waveform (if horizontal units are not set to time). Use the [CURSor:VBArS:UNItS](#) command to specify units. The position can appear in units of base or 1/base.

This command is the equivalent of selecting Cursor Setup from the Cursors menu, selecting V Bars Cursor Type, and then viewing or editing the desired cursor position.

**Group** Cursor

<b>Syntax</b>	<code>CURSOR:VBARS:POS&lt;x&gt; &lt;NR3&gt;</code> <code>CURSOR:VBARS:POS&lt;x&gt;?</code>
<b>Related Commands</b>	<a href="#">CURSOR:VBARS:UNITs</a>
<b>Arguments</b>	<code>&lt;NR3&gt;</code> specifies the cursor position.
<b>Returns</b>	A floating point value with an exponent.
<b>Examples</b>	<code>CURSOR:VBARS:POS2 9.00E-6</code> positions the cursor2 vertical bar cursor at 9 ms. <code>CURSOR:VBARS:POS1?</code> this command might return <code>:CURSOR:VBARS:POS1 1.0000E-06</code> , indicating that the cursor1 vertical bar is positioned at 1 ms.

## CURSOR:VBARS:UNITs

This command sets or queries the units for the vertical bar cursors.

<b>Group</b>	Cursor
<b>Syntax</b>	<code>CURSOR:VBARS:UNITs {SECONds HERtz}</code> <code>CURSOR:VBARS:UNITs?</code>
<b>Arguments</b>	<code>SECONds</code> sets the units of the vertical bar cursors for the time domain (seconds). <code>HERtz</code> sets the units of the vertical bar cursors for the frequency domain (Hertz).
<b>Returns</b>	A string for <code>SECONds</code> or <code>HERtz</code> , depending on the current vertical bar cursor units.
<b>Examples</b>	<code>CURSOR:VBARS:UNITSHERTZ</code> sets the units of the vertical bar cursors to 1/seconds. <code>CURSOR:VBARS:UNITs?</code> might return <code>:CURSOR:VBARS:UNITs SECONds</code> , indicating that the units for the vertical bar cursor are currently set to seconds.

## CURSOR:WAVEform

This command sets or queries the current settings for the waveform cursors or forces them to snap to positions specified by the [DATA:START](#) and [DATA:STOP](#) commands.

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:WAVEform SNAP CURSOR:WAVEform?
<b>Related Commands</b>	<a href="#">DATA:START</a> , <a href="#">DATA:STOP</a>
<b>Arguments</b>	SNAP forces the position of waveform cursor 1 and 2 to snap to the waveform record points specified by the <a href="#">DATA:START</a> and <a href="#">DATA:STOP</a> commands, respectively.
<b>Examples</b>	CURSOR:WAVEFORMSNAP forces waveform cursor1 to the DATA:START position and waveform cursor 2 to the DATA:STOP position.  CURSOR:WAVEFORM? might return the following waveform cursor settings :CURSOR:WAVEFORM:UNITS SECONDS;POSITION1 1.0000E-06;POSITION2 9.0000E-06.

## CURSOR:WAVEform:HDELTA? (Query Only)

This query-only command returns the horizontal difference between the waveform cursors. This is the absolute value of the horizontal position of the first cursor minus the horizontal position of the second cursor. This is equivalent to the waveform delta-time readout value.

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:WAVEform:HDELTA?
<b>Examples</b>	CURSOR:WAVEFORM:HDELTA? might return :CURSOR:WAVEFORM:HDELTA 1.03E-9, indicating that the time between the waveform cursors is 1.03 ns.

## CURSOR:WAVEform:HPOS<x>? (Query Only)

This query-only command returns the position of the specified waveform cursor. The cursor is specified by x, which can be 1 or 2. This corresponds to the v1 or v2 (cursor 1 or cursor 2) cursor readout.

<b>Group</b>	Cursor
--------------	--------

**Syntax** CURSor:WAVEform:HPOS<x>?

**Examples** CURSor:WAVEform:HPOS1? might return :CURSOR:WAVEFORM:HPOS2 0.247344970703, indicating that cursor 2 is at 247.3 mV with respect to ground on the source waveform.

## CURSor:WAVEform:POStion<x>

This command sets or queries the position of a waveform cursor, which is constrained to be visible in the selected time base. (Use the [CURSor:FUNcTion](#) command to set the cursor function to Waveform.) The cursor is specified by x, which can be 1 or 2. This is equivalent to setting a value in the Position control in the Cursor1 or Cursor2 sections of the Cursor Setup dialog box (when Waveform is selected in the Function section).

**Group** Cursor

**Syntax** CURSor:WAVEform:POStion<x> {<NR3>}  
CURSor:WAVEform:POStion<x>?

**Related Commands** [CURSor:FUNcTion](#)

**Arguments** <NR3> specifies the cursor position measured relative to the time of the trigger point of the source waveform.

**Examples** CURSor:WAVEform:POStion2 9.00E-6 sets the waveform cursor 2 position to 9  $\mu$ s.

CURSor:WAVEform:POStion2? might return :CURSOR:WAVEFORM:POSITION2 51.0701402806E-009, indicating waveform cursor 2 is at 51.1 ns.

## CURSor:WAVEform:STYle

This command sets or queries the cursor type for Waveform mode.

**Group** Cursor

**Syntax** CURSor:WAVEform:STYle {LINE\_X|LINES|X}  
CURSor:WAVEform:STYle?

**Arguments**    `LINE_X` specifies the cursor style to be a line with superimposed X.  
                   `LINES` specifies the cursor style to be a line.  
                   `X` specifies the cursor style to be an X.

**Examples**      `CURSOR:WAVEFORM:STYLELINES` sets the cursor style to lines.  
                   `CURSOR:WAVEFORM:STYLE?` might return `:CURSOR:WAVEFORM:STYLE X`,  
                   indicating that the style for waveform cursors is X.

## CURSor:WAVEform:UNIts

This command sets or queries the units for waveform cursors.

**Group**          Cursor

**Syntax**        `CURSor:WAVEform:UNIts BASE`  
                   `CURSor:WAVEform:UNIts?`

**Arguments**    `BASE` sets cursor units to base.

**Examples**      `CURSOR:WAVEFORM:UNITSBASE` sets the waveform cursor units to base.  
                   `CURSOR:WAVEFORM:UNITS?` might return `:CURSOR:WAVEFORM:UNITS BASE`,  
                   indicating that the units for waveform cursors are base.

## CURSor:WAVEform:VDELTA? (Query Only)

This query-only command returns the vertical difference between the waveform cursors. This is the absolute value of the vertical position of the first cursor minus the vertical position of the second cursor. This is equivalent to the waveform delta-voltage readout value.

**Group**          Cursor

**Syntax**        `CURSor:WAVEform:VDELTA?`

**Examples**      `CURSOR:WAVEFORM:VDELTA?` might return `:CURSOR:WAVEFORM:VDELTA`  
                   `1.06E-3`, indicating that the voltage between the waveform cursors is 1.06 mV.

## CURSor:XY? (Query Only)

This query-only command returns all of the XY cursor parameters.

---

**NOTE.** *The values returned for all XY cursor readouts, that do not pertain to the current CURSor:XY:READOUT setting, are returned as the IEEE representation of NAN (Not a Number): 99.1000E+36.*

---

**Group** Cursor

**Syntax** CURSor:XY?

**Examples** CURSOR:XY? might return :CURSOR:XY:PRODELTA  
 99.1000E+36;PRODUCT1 99.1000E+36;PRODUCT2 99.1000E+36;READOUT  
 RECTANGULAR;RADIUS1 99.1000E+36;RADIUS2 99.1000E+36;RATDELTA  
 99.1000E+36;RATIO 199.1000E+36;RATIO2 99.1000E+36;RDELTA  
 99.1000E+36;RECTX1 1.0000;RECTX2 1.0200;RECTY1  
 360.0001E-6;RECTY2360.0001E-6; THDELTA 99.1000E+36;THETA1  
 99.1000E+36;THETA2 99.1000E+36;XDELTA 20.0000E-3;YDELTA 0.0000.

## CURSor:XY:PRODELta? (Query Only)

This query-only command returns the product of the difference between the cursors X positions and Y positions when the CURSor:XY:READOUT is set to PRODUct. The  $\Delta X \times \Delta Y$  value is calculated as (Cursor 2 X - Cursor 1 X)  $\times$  (Cursor 2 Y - Cursor 1 Y).

---

**NOTE.** *This query is only valid when the CURSor:XY:READOUT is set to PRODUct. If the readout is set other than product, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.*

---

**Group** Cursor

**Syntax** CURSor:XY:PRODELta?

**Examples** CURSOR:XY:PRODELTA? might return :CURSOR:XY:PRODELTA -24.9856,  
 indicating that the  $\Delta X \times \Delta Y$  value is -24.9856 VV.



## CURSor:XY:PRODUCT<x>? (Query Only)

This query-only command returns the product of the X and Y positions for the specified cursor when the CURSor:XY:READOUT is set to PRODUct. The cursor is specified by x, which can be 1 or 2.

---

**NOTE.** This query is only valid when the CURSor:XY:READOUT is set to PRODUct. If the readout is set other than product, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:XY:PRODUCT<x>?
<b>Examples</b>	CURSor:XY:PRODUCT1? might return :CURSOR:XY:PRODUCT1 806.4004E-3, indicating that the product of the X position times the Y position for Cursor 1 is 806.4004 mVV.

## CURSor:XY:RADIUS<x>? (Query Only)

This query-only command returns the radius of the specified cursor when the CURSor:XY:READOUT is set to POLARCoord. The cursor is specified by x, which can be 1 or 2.

---

**NOTE.** This query is only valid when the CURSor:XY:READOUT is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:XY:RADIUS<x>?
<b>Examples</b>	CURSor:XY:RADIUS2? might return :CURSOR:XY:RADIUS2 4.9600, indicating that the Cursor 2 radius is 4.9600 V.

## CURSor:XY:RATDELta? (Query Only)

This query-only command returns ratio of the difference between the cursors X position and Y position ( $\Delta Y$ ,  $\Delta X$ ) when the CURSor:XY:READOUT is set

to RATio. The ratio is calculated as  $(\text{Cursor 2 Y} - \text{Cursor 1 Y}) \div (\text{Cursor 2 X} - \text{Cursor 1 X})$ .

---

**NOTE.** This query is only valid when the `CURSOR:XY:READOUT` is set to RATio. If the readout is set other than ratio, the IEEE representation of NAN (Not a Number) is returned: `99.1000E+36`.

---

<b>Group</b>	Cursor
<b>Syntax</b>	<code>CURSOR:XY:RATDELta?</code>
<b>Examples</b>	<code>CURSOR:XY:RATDELTA?</code> might return <code>:CURSOR:XY:RATDELTA -1.0167</code> , indicating that the $\Delta Y/\Delta X$ ratio is -1.0167.

## CURSOR:XY:RATIO<x>? (Query Only)

This query-only command returns ratio the of the X (horizontal) and Y (vertical) position for the specified cursor when the `CURSOR:XY:READOUT` is set to RATio. The cursor is specified by x, which can be 1 or 2. The ratio is calculated as  $Y \div X$ .

---

**NOTE.** This query is only valid when the `CURSOR:XY:READOUT` is set to RATio. If the readout is set other than ratio, the IEEE representation of NAN (Not a Number) is returned: `99.1000E+36`.

---

<b>Group</b>	Cursor
<b>Syntax</b>	<code>CURSOR:XY:RATIO&lt;x&gt;?</code>
<b>Examples</b>	<code>CURSOR:XY:RATIO1?</code> might return <code>:CURSOR:XY:RATIO1 15.8730E-3</code> indicating that the ratio of the X and Y positions for Cursor 1 is 15.8730E-3.

## CURSOR:XY:RDELta? (Query Only)

This query-only command returns the delta radius ( $\Delta r$ ) value when the `CURSOR:XY:READOUT` is set to POLARCoord.

---

**NOTE.** This query is only valid when the `CURSOR:XY:READOUT` is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: `99.1000E+36`.

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:XY:RDELta?
<b>Examples</b>	CURSOR:XY:RDELTA? might return :CURSOR:XY:RDELTA 6.9600, indicating that the difference between the cursors radius is 6.9600 V.

## CURSOR:XY:READOUT

This command sets or queries the XY cursor readout (available when DISPLAY:FORMAT is set to XY).

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:XY:READOUT {RECTangular POLARCoord PRODUCT RATio} CURSOR:XY:READOUT?
<b>Related Commands</b>	<a href="#">DISplay:FORMat</a>
<b>Arguments</b>	<p>RECTangular displays the following values: X1, X2, ΔX, Y1, Y2, ΔY, t1, t2, Δt.</p> <p>POLARCoord displays the following values: r1, r2, Δr, q1, q2, Δq, t1, t2, Δt.</p> <p>PRODUCT displays the following values: X1 x Y1, X2 x Y2, ΔX x ΔY, t1, t2, ΔDt.</p> <p>RATio displays the following values: Y1 , X1, Y2 , X2, ΔY , x ΔX, t1, t2, Δt.</p>
<b>Examples</b>	<p>CURSOR:XY:READOUTRECTangular sets the XY cursor readout to rectangular.</p> <p>CURSOR:XY:READOUT? might return :CURSOR:XY:READOUT RECTANGULAR, indicating that the XY cursor readout is set to rectangular mode.</p>

## CURSOR:XY:RECTX<x>

This command sets or queries the X cursor position in rectangular coordinates. The cursor is specified by x, which can be 1 or 2.

---

**NOTE.** This query is only valid when the CURSOR:XY:READOUT is set to RECTangular. If the readout is set other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:XY:RECTX<x> {<NR3>} CURSOR:XY:RECTX<x>?
<b>Arguments</b>	<NR3> is the X position of the specified cursor in rectangular coordinates.
<b>Examples</b>	CURSOR:XY:RECTX2 5.04 sets the X position of Cursor 2 to 5.04 V in rectangular coordinates.  CURSOR:XY:RECTX1? might return :CURSOR:XY:RECTX1 5.1200, indicating that the X position of Cursor 1 is set to 5.1200 in rectangular coordinates.

## CURSOR:XY:RECTY<x>

This command sets or queries the Y cursor position in rectangular coordinates. The cursor is specified by x, which can be 1 or 2.

---

**NOTE.** This query is only valid when the *CURSOR:XY:READOUT* is set to *RECTangular*. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSOR:XY:RECTY<x> {<NR3>} CURSOR:XY:RECTY<x>?
<b>Arguments</b>	<NR3> is the Y position of the specified cursor in rectangular coordinates.
<b>Examples</b>	CURSOR:XY:RECTY1 2.5 sets the Y position of Cursor 1 to 2.5 V.  CURSOR:XY:RECTY2? might return :CURSOR:XY:RECTY2 4.8800, indicating that the position of Cursor 2 in rectangular coordinates is 4.88 V.

## CURSOR:XY:THDELTA? (Query Only)

This query-only command returns the XY cursor angle delta in polar coordinates.

**NOTE.** This query is only valid when the *CURSor:XY:READOUT* is set to *POLARCoord*. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

- Group** Cursor
- Syntax** CURSor:XY:THDELta?
- Examples** CURSOR:XY:THDELTA? might return :CURSOR:XY:THDELTA 136.3748, indicating that the Dq XY cursor angle is 136.3748 degrees.

### CURSor:XY:THETA<x>? (Query Only)

This query-only command returns the XY cursor angle in polar coordinates. The cursor is specified by x, which can be 1 or 2.

**NOTE.** This query is only valid when the *CURSor:XY:READOUT* is set to *POLARCoord*. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

- Group** Cursor
- Syntax** CURSor:XY:THETA<x>?
- Examples** CURSOR:XY:THETA2? might return :CURSOR:XY:THETA2 -89.0906, indicating that the Channel 2 XY cursor angle in polar coordinates is -89.0906 degrees.

### CURSor:XY:XDELta? (Query Only)

This query-only command returns the XY cursor X delta value in rectangular coordinates.

**NOTE.** This query is only valid when the *CURSor:XY:READOUT* is set to *RECTangular*. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

- Group** Cursor

**Syntax** CURSor:XY:XDELta?

**Examples** CURSOR:XY:XDELTA? might return :CURSOR:XY:XDELTA 40.0000E-3, indicating that the DX value is 40 mV.

## CURSor:XY:YDELta? (Query Only)

This query-only command returns the XY cursor Y delta value in rectangular coordinates.

---

**NOTE.** This query is only valid when the *CURSOR:XY:READOUT* is set to *RECTangular*. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

---

**Group** Cursor

**Syntax** CURSor:XY:YDELta?

**Examples** CURSOR:XY:YDELTA? might return :CURSOR:XY:YDELTA 40.0000E-6, indicating that the DY value is 40  $\mu$ V.

## CURVe

This command transfers waveform data to and 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.

The CURVe command transfers waveform data to the instrument. The data is stored in the reference memory location specified by [DATA:DESTination](#), starting with the data point specified by [DATA:STARt](#). Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

**Group** Waveform Transfer

**Syntax** CURVe  
 CURVe {<Block>|<asc curve>}  
 CURVe?

**Related Commands** DATA:DESTination, DATA:SOURce, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMInpre?, WFMInpre:BYT\_Nr, WFMOutpre?, WFMInpre:NR\_FR, WFMOutpre:NR\_FR

**Arguments** <Block> is the waveform data in binary format. The waveform is formatted as: #<x><yyy><data><newline>, where:

<x> is the number of y bytes. For example, if <yyy>=500, then <x>=3)

---

**NOTE.** <x> is hexadecimal format. The letters A-F denote a number of y bytes between 10 and 15 digits.

---

<yyy> is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. If width is 4, then all bytes on the bus are 4-byte pairs. Use the WFMInpre:BYT\_Nr command to set the width.

<data> is the curve data.

<newline> is a single byte new line character at the end of the data.

<asc curve> is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>...], where each <NR1> represents a data point.

**Examples** CURVE <Block> sets the format of the waveform data, transferred to and from the instrument, to binary format.

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 has been 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 may need increase your program's time-out value to ensure that all data is transferred and read.

---

## CURVNext? (Query Only)

This query-only command returns unique waveform data from the instrument. This query performs just like [CURVe?](#), except multiple uses guarantee that the waveform returned is always a new acquisition since the previous [CURVNext?](#). Note that if the instrument is acquiring waveform records at a slow rate (high resolution mode), you must configure the controller for long timeout thresholds. Data will not be transferred until a new waveform has been acquired since the previous [:CURVNext?](#) response.

---

**NOTE.** *This command does not work with Single Sequence.*

---

**Group** Waveform Transfer

**Syntax** CURVNext?

**Related Commands** [CURVe?](#), [DATA:DESTination](#), [DATA:SOURce](#), [DATA:STARt](#), [DATA:STOP](#), [WFMOutpre?](#)

**Examples** CURVENEXT? This query with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return [:CURVENEXT 61, 62, 61, 60, 60, -59, -59, -58, -58, -59](#).

## CURVEStream

This query continuously transfers waveform data from the instrument as it is acquired. This command puts the instrument into a talk-only mode, allowing the controller to receive waveform records as fast as (and as soon as) they are acquired. Use the [DATA:SOURce](#) command to specify the waveform sources. The command does the same thing as the [CURVe](#) command.



- Control of the instrument through the user interface or other external client is not possible while in streaming mode.
- The GPIB controller must take the instrument out of this continuous talking 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 curve mode: send a device clear over the bus or send another query to the instrument (a MEPE Query Interrupted error will occur, but the instrument will be placed back into its normal talk/listen mode).
- Turning the waveform screen display mode off (:DISPlay:WAVEform OFF) will increase waveform throughput during streaming mode.
- While in streaming mode, two extreme conditions can occur. If the waveform records are being acquired slowly (high resolution), configure the controller for long time-out thresholds, 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.

**NOTE.** *This command does not work with Single Sequence.*

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	CURVEstream {<Block> <asc curve>} CURVEstream?
<b>Related Commands</b>	CURVE, DATA:DESTination, DATA:SOURce, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMInpre?, WFMInpre:BYT_Nr, WFMOutpre?, WFMInpre:NR_FR, WFMOutpre:NR_FR?
<b>Arguments</b>	See the CURVE command.
<b>Returns</b>	Waveform data in its native binary format. To save processing time, DATA:ENCdg FASTEST is set and waveform data is shipped in its raw native format.

## DATA

This command sets or queries the format and location of the waveform data that is transferred with the CURVE command.

**Group** Waveform Transfer

<b>Syntax</b>	DATA {INIT SNAP} DATA?
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA:START</a> , <a href="#">DATA:STOP</a> , <a href="#">DATA:ENCdg</a> , <a href="#">SAVE:WAVEform:FILEFormat</a> , <a href="#">WFMInpre:NR_FR</a> , <a href="#">WFMInpre:NR_Pt</a> , <a href="#">WFMOutpre:NR_FR?</a> , <a href="#">WFMOutpre:NR_Pt?</a>
<b>Arguments</b>	INIT initializes the waveform data parameters to their factory defaults.  SNAP sets DATA:START and DATA:STOP to match the current V Bar/Screen/Waveform cursor positions.
<b>Examples</b>	DATA INIT initializes the waveform data parameters to their factory defaults.  DATA? might return :DATA:DESTINATION REF1; ENCDG RIBINARY; SOURCE CH1; START 1; STOP 1000; FRAMESTART 1; FRAMESTOP 1000.

## DATA:DESTINATION

This command sets or queries the reference memory location for storing waveform data that is transferred into the instrument by the [CURVe](#) command.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	DATA:DESTINATION REF<x> DATA:DESTINATION?
<b>Related Commands</b>	<a href="#">CURVe</a>
<b>Arguments</b>	REF<x> is the reference where the waveform will be stored. The reference number is specified by x, which ranges from 1 through 4.
<b>Examples</b>	DATA:DESTINATION? might return :DATA:DESTINATION REF3, indicating that reference 3 is the currently selected reference memory location for waveform data. DATA:DESTINATION REF1 indicates that incoming waveform data be stored in reference 1.

## 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:ENCdg value causes the corresponding WFMOutpre values to be updated and visa 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|FASTest|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.

- **FASTest** specifies that the data be sent in the fastest possible manner consistent with maintaining accuracy and is interpreted with respect to the first waveform specified in the DATA:SOURce list.

:ENCdg will always be BIN, :BYT\_Or will always be LSB, but :BN\_Fmt and :BYT\_Nr will depend on the first DATA:SOURce waveform. :BN\_Fmt will be RI unless the waveform is internally stored as a floating point number, in which case the FP format will be used.

- **RIBinary** specifies signed integer data point representation with the most significant byte transferred first.

When :BYT\_Nr is 1, the range is from -128 through 127. When :BYT\_Nr is 2, the range is from -32,768 through 32,767. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. Center screen is 0 (zero). The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen. This is the default argument.

- **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. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. The center of the screen is 127. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

- **FPBinary** specifies the floating point (width = 4) data.

The range is from  $-3.4 \times 10^{38}$  to  $3.4 \times 10^{38}$ . The center of the screen is 0. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

The **FPBinary** argument is only applicable to math waveforms or ref waveforms saved from math waveforms.

- **SRIBinary** is the same as **RIBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

- **SRPBinary** is the same as **RPBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

- **SFPBinary** specifies floating point data in IBM PC format. The **SFPBinary** argument only works on math waveforms or ref waveforms saved from math waveforms.

**Table 2-42: DATA and WFMOupre Parameter Settings**

DATA:ENCdg Setting	WFMOupre Settings			
	:ENCdg	:BN_Fmt	:BYT_Or	:BYT_NR
AScii	ASC	N/A	N/A	1,2,4,8
FAStest	BIN	RI/FP	MSB	1,2,4
RIBinary	BIN	RI	MSB	1,2,8
RPBinary	BIN	RP	MSB	1,2,8
FPBinary	BIN	FP	MSB	4
SRIbinary	BIN	RI	LSB	1,2,8
SRPbinary	BIN	RP	LSB	1,2,8
SFPbinary	BIN	FP	LSB	4

**Examples** DATA:ENCDGRPBinary sets the data encoding format to be a positive integer where the most significant byte is transferred first.

DATA:ENCDG? might return :DATA:ENCDG SRPBINARY for the format of the outgoing waveform data.

## DATA:FRAMESTART

This command sets or queries the starting data frame for waveform transfer. This command allows for transferring partial frames of the waveform to and from the instrument.

**Group** Waveform Transfer

**Syntax** DATA:FRAMESTART <NR1>  
DATA:FRAMESTART?

**Arguments** <NR1> is the first data frame that is transferred, which ranges from 1 to the number of acquired frames. Data is transferred from <NR1> to [DATA:FRAMESTOP](#) or the number a acquired frames, whichever is less. If <NR1> is greater than the number of acquired frames, then you will receive the last acquired frame. If no frames were acquired, then you receive the first frame which contains all nulls.

**Returns** The starting data frame for waveform transfer.

**Examples** [DATA:FRAMESTART](#) 10 specifies that the waveform transfer will begin with frame 10.

[DATA:FRAMESTART?](#) might return `:DATA:FRAMESTART 214` indicating the first waveform frame that will be transferred.

## DATA:FRAMESTOP

This command sets or queries the ending data frame for waveform transfer. This command allows for transferring partial frames of the waveform to and from the instrument.

**Group** Waveform Transfer

**Syntax** `DATA:FRAMESTOP <NR1>`  
`DATA:FRAMESTOP?`

**Arguments** `<NR1>` is the last data frame that is transferred, which ranges from 1 to the number of acquired frames. If `<NR1>` is greater than the number of acquired frames, then you will receive up to the number of acquired frames. If both `DATA:FRAMESTART` and `DATA:FRAMESTOP` are greater than the number of acquired frames, then you will receive the last acquired frame. If no frames were acquired, then you receive the first frame which contains all nulls.

`DATA:FRAMESTART` and `DATA:FRAMESTOP` are order independent. The smaller number is used as in [CURVe](#) and [SAVE:WAVEform](#) queries.

To always transfer complete waveforms, set `DATA:FRAMESTART` to 1 and `DATA:FRAMESTOP` to the maximum number of acquired frames, or larger.

**Returns** The last data frame that will be transferred.

**Examples** [DATA:FRAMESTOP 1500](#) specifies that the waveform transfer will stop with frame 1500.

[DATA:FRAMESTOP?](#) might return `:DATA:FRAMESTOP 14900` indicating the last waveform frame that will be transferred.

## 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 <wfm> [<,><wfm>]  
DATA:SOURCE?

**Related Commands** [CURVe](#)

**Arguments** <wfm> is the location of the waveform data that will be transferred from the instrument to the controller.

**Examples** DATA:SOURCECH1 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: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](#), [SAVE:WAVEform](#), [SAVE:WAVEform:FILEFormat](#), [WFMinpre:NR\\_Pt](#), [WFMOupre:NR\\_Pt?](#), [WFMinpre:NR\\_FR](#), [WFMOupre:NR\\_FR?](#)

**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:START10 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 and [WFMinpre:NR\\_Pt](#) is used.

---

**Group** Waveform Transfer

**Syntax** DATA:STOP <NR1>  
DATA:STOP?

**Related Commands** [CURVe](#), [DATA](#), [DATA:START](#), [SAVE:WAVEform](#), [SAVE:WAVEform:FILEFormat](#), [WFMinpre:NR\\_Pt](#), [WFMOupre:NR\\_Pt?](#), [WFMinpre:NR\\_FR](#), [WFMOupre:NR\\_FR?](#)

**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:STOP15000 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:SYNCSOURces

This command sets or queries if the data sync sources are on or off.



<b>Group</b>	Waveform Transfer
<b>Syntax</b>	DATA:SYNCSOURCES {<NR1> OFF ON} DATA:SYNCSOURCES?
<b>Arguments</b>	NR1 = 0 specifies that the data sync sources are not available; any other value specifies that the data sync sources are available.  OFF specifies that the data sync sources are not available.  ON specifies that the data sync sources are available.
<b>Examples</b>	DATA:SYNCSOURCES ON sets the data sync sources are available.  DATA:SYNCSOURCES? might return DATA:SYNCSOURCES OFF, indicating that the data sync sources are not available.

## DATE

This command sets or queries the date that the instrument can display. This command is equivalent to selecting Set Date & Time from the Utilities menu and then setting the fields in the Date group box.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	DATE <QString> DATE?
<b>Related Commands</b>	<a href="#">TIME</a>
<b>Arguments</b>	<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.
<b>Examples</b>	DATE"2000-01-24" specifies that the date is set to January 24, 2000.  DATE? might return :DATE 2000-01-24, indicating the current date is set to January 24, 2000.

## \*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a \*TRG command or the GET GPIB interface message. 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 #OACQUIRE:STATE RUN specifies that the acquisition system will be started each time a \*TRG command is sent.

## DELEte:SETUp (No Query Form)

This command (no query form) changes the setup to reference the factory setup instead of the specific user setup slot. The content of the setup slot is unchanged, but the data will no longer be accessible to you.

This command is equivalent to selecting Delete from the File menu, and then clicking the specific setup you want to delete (user setups are shown as Setup-User and are ordered in the list from 1 through 10, if defined) or All Setups.

---

**NOTE.** *The setup information cannot be recovered once it has been deleted.*

---

**Group** Save and Recall

**Syntax**     `DELEte:SETUp {ALL|<NR1>}`

**Related Commands**     [\\*RCL](#), [RECAIl:SETUp](#), [\\*RST](#), [\\*SAV](#), [SAVe:SETUp](#), [\\*SDS](#)

**Arguments**     ALL deletes all the stored setups.  
                        <NR1> specifies a setup storage location to delete. Setup storage location values range from 1 through 10; using an out-of-range value causes an error.

**Examples**     `DELETE:SETUP ALL` deletes all stored setups. All ten storage locations are initialized to the factory default setup.

## DELEte:WAVEform (No Query Form)

This command (no query form) deletes one or all stored reference waveforms from memory.

This command is equivalent to selecting Delete from the File menu, and then choosing the reference waveform you want to delete; choosing All Refs deletes all of the reference waveforms.

**Group**     Save and Recall

**Syntax**     `DELEte:WAVEform {ALL|REF<x>}`

**Related Commands**     [RECAIl:WAVEform](#), [SAVe:WAVEform](#)

**Arguments**     ALL specifies to delete all the stored reference waveforms.  
                        REF<x> specifies to delete one of the reference memory locations. Reference memory location values range from 1 through 4.

**Examples**     `DELETE:WAVEFORM ALL` removes all waveforms stored in reference memory.  
                        `DELETE:WAVEFORM REF2` removes the waveform stored at REF2.

## 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** DESE209 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:CONTROL:HALT

This command determines or queries whether the next execution of diagnostics looping will stop on the first diagnostic failure that occurs or will continue to loop on the selected set of diagnostic functions. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then enabling Halt on Fail.

**Group** Diagnostics

**Syntax** DIAG:CONTROL:HALT {ON|OFF|<NR1>}  
DIAG:CONTROL:HALT?

**Related Commands** [DIAG:CONTROL:LOOP](#)

**Arguments** ON enables the halt function, causing the execution of diagnostics looping to halt at the first diagnostic failure that occurs.

OFF disables the halt looping function, allowing the instrument to continue to loop on the entire set of diagnostics, even if diagnostic failure occurs.

<NR1> = 0 enables the halt function; any other value disables the halt function.

**Examples** DIAG:CONTROL:HALT ON enables the halt function, causing the execution of diagnostics looping to halt at the first diagnostic failure that occurs.

DIAG:CONTROL:HALT? might return :DIAG:CONTROL:HALT 0, indicating that the halt function is enabled.

## DIAG:CONTROL:LOOP

This command determines or queries whether the next execution of diagnostics executes once or continuously loops on the selected set of diagnostics (assuming the halt control is set to off using the [DIAG:CONTROL:HALT](#) command or that the halt control is set to ON but no failures occur). This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then enabling Loop Control.

**Group** Diagnostics

**Syntax** DIAG:CONTROL:LOOP {ON|OFF|<NR1>}  
DIAG:CONTROL:LOOP?

**Related Commands** [DIAG:CONTROL:HALT](#), [DIAG:STOP](#)

**Arguments** ON enables the loop function, causing the execution of diagnostics to continuously loop.

OFF disables the loop function, causing the instrument to execute the entire set of diagnostics once and then halt.

<NR1> = 0 enables the loop function; any other value disables the loop function.

**Examples** DIAG:CONTROL:LOOP ON enables the loop function.

DIAG:CONTROL:LOOP? might return :DIAG:CONTROL:LOOP 0, indicating that looping on diagnostics is enabled.

## DIAG:EXECUTE (No Query Form)

This command (no query form) starts the execution of the currently selected set of diagnostics. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then pressing Run.

**Group** Diagnostics

**Syntax** DIAG:EXECUTE

**Related Commands** [DIAG:STATE](#)

**Examples** DIAG:EXECUTE starts the execution of the entire set of diagnostics.

## DIAG:FAILURES:CLEAR

This command sets and queries the clearing of pass/fail information from data structures, not the Event Log, at the start of diagnostic tests.

**Group** Diagnostics

**Syntax** DIAG:FAILURES:CLEAR {<NR1>|OFF|ON}  
DIAG:FAILURES:CLEAR?

**Arguments** <NR1> = 0 turns off the clearing the pass/fail information at the start of tests; any other value turns on the clearing of pass/fail information.

OFF does not clear pass/fail information at the start of tests.

ON clears pass/fail information from data structures, not the Event Log, at the start of tests.

**Examples** DIAG:FAILURES:CLEARON clears pass/fail information on the start of tests.

DIAG:FAILURES:CLEAR? might return DIAG:FAILURES:CLEAR1 indicating that pass/fail information will be cleared at the start of tests.

## DIAG:ITEM? (Query Only)

This query-only command returns the diagnostics settings. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, and then reviewing the diagnostics settings.

This query-only command can be issued while diagnostics are still in progress.

**Group** Diagnostics

**Syntax** `DIAG:ITEM? <NR1>`

**Related Commands** [DIAG:ITEM:FAILURES?](#), [DIAG:ITEM:NAME?](#), [DIAG:ITEM:RESULT?](#), [DIAG:ITEM:SUBITEMS?](#)

**Arguments** `<NR1>` sets the index item about which data will be returned, which ranges from 0 through 15.

**Examples** `DIAG:ITEM? 2` might return `:DIAG:ITEM 2,"Acquisition","Pass",0,8`, indicating that the acquisition test passed and that there were 0 failures of the 8 subitems for that test.

## DIAG:ITEM:FAILURES? (Query Only)

This query-only command returns the total number of failures. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group** Diagnostics

**Syntax** `DIAG:ITEM:FAILURES? <NR1>`

**Related Commands** [DIAG:ITEM?](#), [DIAG:ITEM:NAME?](#), [DIAG:ITEM:RESULT?](#), [DIAG:ITEM:SUBITEMS?](#)

**Arguments** `<NR1>` sets the index item about which data will be returned, which ranges from 0 through 15.

**Examples**     `DIAG:ITEM:FAILURES? 2` might return `:DIAG:ITEM:FAILURES 2,0`, indicating that there were 0 failures for item 2.

## DIAG:ITEM:NAME? (Query Only)

This query-only command returns the name of the selected menu item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Subsystem, Area and Test settings.

This query-only command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:ITEM:NAME? <NR1>`

**Related Commands**     [DIAG:ITEM?](#), [DIAG:ITEM:FAILURES?](#), [DIAG:ITEM:RESULT?](#), [DIAG:ITEM:SUBITEMS?](#)

**Arguments**     `<NR1>` sets the index item about which data will be returned, which ranges from 0 through 15.

**Examples**     `DIAG:ITEM:NAME? 2` might return `:DIAG:ITEM:NAME 2,"Acquisition"` indicating that the selected menu item (2) is Acquisition.

## DIAG:ITEM:RESULT? (Query Only)

This query-only command returns the result from the last execution of the item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:ITEM:RESULT? <NR1>`

**Related Commands**     [DIAG:ITEM?](#), [DIAG:ITEM:NAME?](#), [DIAG:ITEM:FAILURES?](#), [DIAG:ITEM:SUBITEMS?](#)



**Arguments** <NR1> sets the index item about which data will be returned, which ranges from 0 through 15.

**Examples** `DIAG:ITEM:RESULT? 2` might return `:DIAG:ITEM:RESULT 2,"Pass"`, indicating that the item passed during the last execution.

`DIAG:ITEM:RESULT? 2` might return `:DIAG:ITEM:RESULT 2,"Fail"`, indicating that the item failed during the last execution.

`DIAG:ITEM:RESULT? 2` might return `:DIAG:ITEM:RESULT 2," "`, indicating that the item was not run.

## DIAG:ITEM:SUBITEMS? (Query Only)

This query-only command returns the number of subitems associated with the item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, choosing the Subsystem, Area or Test setting and then reviewing the resulting subitems.

This query-only command can be issued while diagnostics are still in progress.

**Group** Diagnostics

**Syntax** `DIAG:ITEM:SUBITEMS? <NR1>`

**Related Commands** [DIAG:ITEM?](#), [DIAG:ITEM:NAME?](#), [DIAG:ITEM:FAILURES?](#), [DIAG:ITEM:RESULT?](#)

**Arguments** <NR1> sets the index item about which data will be returned, which ranges from 0 through 15.

**Examples** `DIAG:ITEM:SUBITEMS? 2` might return `:DIAG:ITEM:SUBITEMS 2,8`, indicating that there are 8 subitems associated with the selected item.

## DIAG:LEVEL

This command sets or queries the selected level of diagnostic test hierarchy. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

**Group** Diagnostics

**Syntax**     `DIAG:LEVEL {SUBSYS|AREA|TEST}`  
`DIAG:LEVEL?`

**Arguments**     `SUBSYS` sets diagnostic testing to the subsystem level.  
`AREA` sets diagnostic testing to the area level.  
`TEST` sets diagnostic testing to the test level.

**Examples**     `DIAG:LEVELAREA` sets the level of diagnostic test hierarchy to Area.  
`DIAG:LEVEL?` might return `:DIAG:LEVEL SUBSYSI`, indicating that the current level of diagnostic test hierarchy is Subsys.

## DIAG:LOOPS? (Query Only)

This query-only command returns the number of times that the selected diagnostics set was completed during the last diagnostic execution. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Elapsed Loops.

This query-only command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:LOOPS?`

**Examples**     `DIAG:LOOPS?` might return `:DIAG:LOOPS 5` indicating that the selected set of diagnostics was completed five times during the last diagnostic execution.

## DIAG:NAME? (Query Only)

This query-only command returns the names of the subsystem, area, and test of the current diagnostic test. This command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:NAME?`

**Related Commands**     [DIAG:NAME:AREA?](#), [DIAG:NAME:SUBSYS?](#), [DIAG:NAME:TEST?](#)

**Examples**     `DIAG:NAME?` might return `:DIAG:NAME:SUBSYS "Processor";AREA "Memory";TEST"DRAMwalk1"`, indicating the subsystem name, area name, and test name of the currently selected diagnostic test.

## DIAG:NAME:AREA? (Query Only)

This query-only command returns the selected area of the current diagnostic test. There are three levels of diagnostic test hierarchy: subsystem, area and test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:NAME:AREA?`

**Related Commands**     [DIAG:NAME?](#), [DIAG:NAME:SUBSYS?](#), [DIAG:NAME:TEST?](#)

**Examples**     `DIAG:NAME:AREA?` might return `:DIAG:NAME:AREA "Memory"` indicating the selected area name of the current diagnostic test.

## DIAG:NAME:SUBSYS? (Query Only)

This query-only command returns the subsystem of the current diagnostic test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group**     Diagnostics

**Syntax**     `DIAG:NAME:SUBSYS?`

**Related Commands**     [DIAG:NAME?](#), [DIAG:NAME:AREA?](#), [DIAG:NAME:TEST?](#)

**Examples**     `DIAG:NAME:SUBSYS?` might return `:DIAG:NAME:SUBSYS "Acquisition"`, indicating the subsystem name of the current diagnostic test.

## DIAG:NAME:TEST? (Query Only)

This query-only command returns the name of the current diagnostic test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group** Diagnostics

**Syntax** DIAG:NAME:TEST?

**Related Commands** [DIAG:NAME?](#), [DIAG:NAME:AREA?](#), [DIAG:NAME:SUBSYS?](#)

**Examples** DIAG:NAME:TEST? might return :DIAG:NAME:TEST "diagDataFormatConf", indicating the test name of the current diagnostic test.

## DIAG:NUMITEMS? (Query Only)

This query-only command returns the number of items on the currently selected level of test hierarchy, which ranges from 1 through 15. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

**Group** Diagnostics

**Syntax** DIAG:NUMITEMS?

**Examples** DIAG:NUMITEMS? might return :DIAG:NUMITEMS 7, indicating the number of items on the currently selected level of test hierarchy.

## DIAG:RESuLts? (Query Only)

This query-only command returns an abbreviated status about the results of the last diagnostic (or test) execution. For a more explanatory status message, use the DIAG:RESuLts:VERBose? query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

---

<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:RESuLts?
<b>Related Commands</b>	<a href="#">DIAG:RESuLts:VERBoSe?</a>
<b>Examples</b>	<p>DIAG:RESULTS? might return :DIAG:RESULTS "135", indicating the specific failure code of the first test failure that occurred.</p> <p>DIAG:RESULTS? might return :DIAG:RESULTS "Pass", indicating that the last diagnostic test passed.</p> <p>DIAG:RESULTS? might return :DIAG:RESULTS "*****", indicating that the diagnostic test was not run.</p>

## DIAG:RESuLts:VERBoSe? (Query Only)

This query-only command returns a more explanatory message about the results of the last diagnostic (or test) execution than the `DIAG:RESuLts?` query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:RESuLts:VERBoSe?
<b>Related Commands</b>	<a href="#">DIAG:RESuLts?</a>
<b>Examples</b>	<p>DIAG:RESULTS:VERBOSE? might return</p> <p>:DIAG:RESULTS:VERBOSE "DIAG ABORTED, ERROR 15 in A2D-Dmux 1 interconnects".</p>

## DIAG:SELEct:ALL (No Query Form)

This command (no query form) selects all available diagnostics. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing ALL from the Subsystem, Area and Test pull-down lists.

<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:SElect:ALL ALL
<b>Related Commands</b>	<a href="#">DIAG:NAME:AREA?</a> , <a href="#">DIAG:NAME:SUBSYS?</a> , <a href="#">DIAG:NAME:TEST?</a> , <a href="#">DIAG:EXECUTE</a>
<b>Arguments</b>	ALL selects all available diagnostics.
<b>Examples</b>	DIAG:SELECT:ALL ALL selects all available diagnostics.

## DIAG:SElect:AREA

This command selects or queries an available diagnostic area. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then selecting an Area from the pull-down list.

---

**NOTE.** This command should be executed only if *DIAG:LEVEL* is currently set to *AREA*.

---

<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:SElect:AREA <NR1> DIAG:SElect:AREA?
<b>Related Commands</b>	<a href="#">DIAG:SElect:SUBSYS</a> , <a href="#">DIAG:SElect:TEST</a> , <a href="#">DIAG:SElect:LAST</a>
<b>Arguments</b>	<NR1> selects a diagnostic area by number, which can range from 0 (zero selects all) through 15 (as specified by <a href="#">DIAG:NUMITEMS?</a> ).
<b>Examples</b>	DIAG:SELECT:AREA 1 selects Area 1 as the area to be diagnosed.  DIAG:SELECT:AREA? might return :DIAG:SELECT:AREA 0, indicating that all areas are selected.

## DIAG:SElect:LAST

This command selects one or more diagnostic menu items to be executed via the [DIAG:EXECUTE](#) command. If you specify *DIAG:LEVEL SUBSYS*, then menu

items come from this diagnostic level and are limited to the value returned by the `DIAG:NUMITEMS?` query. For example, if the return from the `DIAG:NUMITEMS?` query is `:DIAG:NUMITEMS 5`, specifying `DIAG:SELEct:SUBSYS 2` indicates that diagnostics will start from subsystem 2 and that you can specify a range from 2 through 5 for `DIAG:SELEct:LAST`.

If you enter `:DIAG:SELEct:LAST 2`, only subsystem 2 will be executed. `DIAG:SELEct:LAST 4`, subsystems 2 *through* 4 will be executed.

**Group** Diagnostics

**Syntax** `DIAG:SELEct:LAST <NR1>`  
`DIAG:SELEct:LAST?`

**Related Commands** [DIAG:EXECUTE](#), [DIAG:LEVEL](#), [DIAG:NAME:AREA?](#),  
[DIAG:NAME:SUBSYS?](#), [DIAG:NAME:TEST?](#), [DIAG:NUMITEMS?](#),  
[DIAG:SELEct:SUBSYS](#)

**Arguments** `<NR1>` selects an integer that identifies the number of the last item that will be executed when the [DIAG:EXECUTE](#) command is run.

**Examples** `DIAG:SELEct:LAST 2` specifies that (based on the example in the description above) only subsystem 2 will be executed.

`DIAG:SELEct:LAST?` might return `:DIAG:SELEct:LAST 5`, indicating that 5 is the last item of a group of items from the current level of test hierarchy.

## DIAG:SELEct:SUBSYS

This command selects or queries the available subsystem. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing a Subsystem from the drop-down list.

---

**NOTE.** *This command should be executed only if `DIAG:LEVEL` is currently set to `SUBSYS`.*

---

**Group** Diagnostics

**Syntax** `DIAG:SELEct:SUBSYS <NR1>`  
`DIAG:SELEct:SUBSYS?`

**Related Commands**    [DIAg:SElect:AREA](#), [DIAg:SElect:LAST](#)

**Arguments**    <NR1> selects a subsystem by number, which can range from 0 (zero selects ALL) through 15 (as limited by the return from [DIAg:NUMITEMS?](#)).

**Examples**    `DIAG:SELECT:SUBSYS 1` selects Subsystem 1 as the subsystem to be diagnosed.  
`DIAG:SELECT:SUBSYS?` might return `:DIAG:SELECT:SUBSYS 0`, indicating that all subsystems are selected.

## DIAg:SElect:TEST

This command selects or queries one of the available tests. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing a Test from the drop-down list.

---

**NOTE.** *This command should be executed only if `DIAG:LEVEL` is currently set to `TEST`.*

---

**Group**    Diagnostics

**Syntax**    `DIAG:SElect:TEST <NR1>`  
`DIAG:SElect:TEST?`

**Related Commands**    [DIAg:SElect:AREA](#), [DIAg:SElect:LAST](#), [DIAg:SElect:SUBSYS](#)

**Arguments**    <NR1> selects a test by number, which can range from 0 (zero selects ALL) through 15 (as limited by the return from [DIAg:NUMITEMS?](#)).

**Examples**    `DIAG:SELECT:TEST 1` selects Test 1 as the test to be executed.  
`DIAG:SELECT:TEST?` might return `:DIAG:SELECT:TEST 0`, indicating all tests have been selected.

## DIAg:STATE

This command changes the instrument operating state. Depending on the argument, diagnostics capabilities are either turned on or off. This command is equivalent to opening the `DIAG:STATE` dialog (ON) or closing it (OFF).



<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:STATE {EXECUTE ON OFF <NR1>} DIAG:STATE?
<b>Related Commands</b>	<a href="#">TEST:STOP</a>
<b>Arguments</b>	<p>EXECUTE starts execution of the diagnostics.</p> <p>ON puts the instrument into the state in which diagnostics can be run. This argument is thrown automatically if the <a href="#">DIAG:EXECUTE</a> command is executed.</p> <p>OFF disables diagnostics capabilities and returns the instrument to a normal operating state.</p> <p>&lt;NR1&gt; = 0 disables diagnostics capabilities and returns the instrument to a normal operating state; any other value enables diagnostics.</p>
<b>Examples</b>	<p>DIAG:STATE OFF turns off diagnostics capabilities and returns the instrument to a normal operating state.</p> <p>DIAG:STATE? might return :DIAG:STATE 0, indicating that diagnostics are disabled.</p>

## DIAG:STOP (No Query Form)

This command (no query form) causes diagnostics (or test) execution to terminate at the end of the next low-level test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking Abort.

<b>Group</b>	Diagnostics
<b>Syntax</b>	DIAG:STOP
<b>Related Commands</b>	<a href="#">TEST:STOP</a>
<b>Examples</b>	DIAG:STOP terminates diagnostics (or test) execution at the end of the next low-level test.

## DISplay? (Query Only)

This query-only command returns the current Display settings. This command is equivalent to selecting Display Setup from the Display menu and then viewing the contents of each tab.

**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:CLOCK

This command sets or queries the display of the date and time. This is equivalent to selecting Display Date & Time from the Display menu. The query form of this command returns an ON (1) or an OFF (0).

**Group** Display Control

**Syntax** DISplay:CLOCK {ON|OFF|<NR1>}  
 DISplay:CLOCK?

**Related Commands**    [DATE](#), [TIME](#)

**Arguments**    ON enables display of date and time.  
 OFF disables display of date and time.  
 <NR1> = 0 disables display of date and time; any other value enables display of date and time.

**Examples**    DISPLAY:CLOCKON enables display of date and time.  
 DISPLAY:CLOCK? might return :DISPLAY:CLOCK 1 indicating that the display of date and time is currently enabled.

## DISplay:COLOr? (Query Only)

This query-only command returns the settings from the PALETTE, MATHCOLOR and REFCOLOR commands. This is equivalent to selecting Colors from the Display menu.

**Group**    Display Control

**Syntax**    DISplay:COLOr?

**Arguments**    None

**Examples**    DISPLAY:COLOR? might return :DISPLAY:COLOR:PALETTE:IMAGEVIEW  
 TEMPERATURE;RECORDVIEW NORMAL;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 indicating that the primary source waveform color is used for each  
 math trace.

## DISplay:COLOr:MATHCOLOR

This command sets or queries the color to be used for math traces, either in the standard palette's nominal Math color, or according to the color of the source waveform. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Colors tab.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOR:MATHCOLOR {DEFAULT INHERIT} DISplay:COLOR:MATHCOLOR?
<b>Related Commands</b>	<a href="#">DISplay:COLOR:PALEtte:IMAGEView</a> , <a href="#">DISplay:COLOR:PALEtte:RECORDView</a>
<b>Arguments</b>	<p>DEFAULT sets color math traces in nominal palette math color, which is red.</p> <p>INHERIT sets color math traces in the source waveform color. Math waveforms are drawn in the same color as their primary source waveform.</p>
<b>Examples</b>	<p>DISPLAY:COLOR:MATHCOLORDEFAULT sets the color math traces in the nominal palette math color.</p> <p>DISPLAY:COLOR:MATHCOLOR? might return :DISPLAY:COLOR:MATHCOLOR INHERIT, indicating that the primary source waveform color is used for each math trace.</p>

## DISplay:COLOR:PALEtte:IMAGEView

This command sets or queries the color palette for all image view (DPO and WfmDB) traces.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOR:PALEtte:IMAGEView {NORMal MONOGREEN MONOGRAY TEMPERature SPECTra1 USER} DISplay:COLOR:PALEtte:IMAGEView?
<b>Arguments</b>	<p>NORMal colors traces according to their channel. This is the default color palette.</p> <p>MONOGREEN colors traces green, emulating a traditional instrument color palette.</p> <p>MONOGRAY colors traces gray, emulating a monochrome instrument.</p> <p>TEMPERature</p> <p>All traces share a multicolored palette, where “intensity” is represented by hue; blue for least frequently hit, red for most frequently hit.</p> <p>SPECTra1.</p>

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

USER.

All traces share a user-defined palette.

### Examples

DISPLAY:COLOR:PALETTE:IMAGEVIEWMONOGRAY sets the palette for all display types.

DISPLAY:COLOR:PALETTE:IMAGEVIEW? might return :DISPLAY:COLOR:PALETTE:IMAGEVIEW NORMAL, indicating that the Normal color palette is currently selected, which colors traces according to their channel.

## DISPlay:COLOR:PALEtte:RECORDView

This command sets or queries the color palette for all record view (non image view) waveforms.

**Group** Display Control

**Syntax** DISPlay:COLOR:PALEtte:RECORDView  
{NORMAL|MONOGREEN|MONOGRAY|TEMPERature|SPECTra1|USER}  
DISPlay:COLOR:PALEtte:RECORDView?

**Arguments** NORMAL colors traces according to their channel. This is the default color palette.  
MONOGREEN colors traces green, emulating a traditional instrument color palette.  
MONOGRAY colors traces gray, emulating a monochrome instrument.  
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.  
SPECTra1 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.  
USER colors all traces using a user-defined palette. All traces share this palette.

### Examples

DISPLAY:COLOR:PALETTE:RECORDVIEWMONOGRAY sets the palette for all display types.

DISPLAY:COLOR:PALETTE:RECORDVIEW? might return :DISPLAY:COLOR:PALETTE:RECORDVIEW NORMAL, indicating that

the Normal color palette is currently selected, which colors traces according to their channel.

## DISplay:COLOr:PALEtte:USER

This command queries the color palette for group settings. It outputs settings from the DISPlay CARET, CH<x>, GRATICULE, HISTOGRAM, MASK, MASKHIGHLIGHT, MATH<x>, and REF<x> commands.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOr:PALEtte:USER RESET DISplay:COLOr:PALEtte:USER?
<b>Arguments</b>	RESET sets all user palettes to their default values.
<b>Examples</b>	DISPLAY:COLOR:PALETTE:USER? might return :DISPLAY:COLOR:PALETTE: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:PALEtte:USER:CAREt

This command sets or queries the caret color for the user palette. It assigns the HUE, light, Saturation (HLS) triplet used for the caret color. The caret is the solid, inverted delta positioned on the top graticule line, which indicates the trigger position within the waveform record.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOr:PALEtte:USER:CAREt <NR1>, <NR1>, <NR1> DISplay:COLOr:PALEtte:USER:CAREt?
<b>Arguments</b>	<NR1> Hue. Range of 0 to 360. <NR1> Lightness. Range of 0 to 100. <NR1> Saturation. Range of 0 to 100.

**Examples**     `DISPLAY:COLOR:PALETTE:USER:CARET330,75,100` sets the caret color hue to 330, the lightness to 75, and the saturation to 100.

`DISPLAY:COLOR:PALETTE:USER:CARET?` might return  
`:DISPLAY:COLOR:PALETTE:USER:CARET 250, 85, 90`, indicating that the color hue is set at 250, the lightness is set to 85, and the saturation is set to 90.

## DISplay:COLOR:PALEtte:USER:CH<x>

This command sets or queries the hue-lightness-saturation (HLS) triplet to be used for the specified channel color for the user palette. The channel is specified by x. The value of x can range from 1 through 4.

**Group**     Display Control

**Syntax**     `DISplay:COLOR:PALEtte:USER:CH<x> <NR1>, <NR1>, <NR1>`  
`DISplay:COLOR:PALEtte:USER:CH<x>?`

**Arguments**     `<NR1>` Hue. Range of 0 to 360.  
                       `<NR1>` Lightness. Range of 0 to 100.  
                       `<NR1>` Saturation. Range of 0 to 100.

**Examples**     `DISplay:COLOR:PALEtte:USER:CH1 330,75,100` sets a user palette of hue = 330, lightness = 75, and saturation = 100 for Channel 1.

`DISplay:COLOR:PALEtte:USER:CH2?` might return  
`:DISPLAY:COLOR:PALETTE:USER:CH2 325,65,100` indicating that Channel 2 color for the user palette is set at 325 for hue, 65 for lightness, and 100 for saturation.

## DISplay:COLOR:PALEtte:USER:GRAticule

This command sets or queries the user palette graticule color. It assigns the hue-lightness-saturation triplet to be used for the graticule color for the user palette.

**Group**     Display Control

**Syntax**     `DISplay:COLOR:PALEtte:USER:GRAticule <NR1>, <NR1>, <NR1>`  
`DISplay:COLOR:PALEtte:USER:GRAticule?`

<b>Arguments</b>	<NR1> Hue. Range of 0 to 360. <NR1> Lightness. Range of 0 to 100. <NR1> Saturation. Range of 0 to 100.
<b>Examples</b>	DISPLAY:COLOR:PALETTE:USER:GRATICULE330,75,100 sets the graticule color for the user palette.  DISPLAY:COLOR:PALETTE:USER:GRATICULE? might return :DISPLAY:COLOR:PALETTE:USER:GRATICULE 300,60,85, indicating that the user palette graticule color is assigned a hue of 300, a lightness of 60, and a saturation of 85.

## DISplay:COLOR:PALEtte:USER:HIStogram

This command sets or queries the user palette histogram color. It assigns the hue-saturation-lightness (HLS) triplet to be used for the histogram color for the user palette.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOR:PALEtte:USER:HIStogram <NR1>,<NR1>,<NR1> DISplay:COLOR:PALEtte:USER:HIStogram?
<b>Arguments</b>	<NR1> Hue. Range of 0 to 360. <NR1> Lightness. Range of 0 to 100. <NR1> Saturation. Range of 0 to 100.
<b>Examples</b>	DISPLAY:COLOR:PALETTE:USER:HISTOGRAM330,75,100 sets the histogram color for the user palette.  DISPLAY:COLOR:PALETTE:USER:HISTOGRAM? might return :DISPLAY:COLOR:PALETTE:USER:HISTOGRAM 330,50,90, indicating that the user palette histogram color is assigned a hue of 330, a lightness of 50, and a saturation of 90.

## DISplay:COLOR:PALEtte:USER:MASK

This command sets or queries the user palette mask color. It assigns the hue-lightness-saturation (HLS) color for the mask color for the user palette.



<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOR:PALETTE:USER:MASK <NR1>,<NR1>,<NR1> DISplay:COLOR:PALETTE:USER:MASK?
<b>Arguments</b>	<NR1> Hue. Range of 0 to 360. <NR1> Lightness. Range of 0 to 100. <NR1> Saturation. Range of 0 to 100.
<b>Examples</b>	DISPLAY:COLOR:PALETTE:USER:MASK330,75,100 sets the mask color for the user palette.  DISPLAY:COLOR:PALETTE:USER:MASK? might return :DISPLAY:COLOR:PALETTE:USER:MASK 330,50,90, indicating that the user palette mask color is assigned a hue of 330, a lightness of 50, and a saturation of 90.

## DISplay:COLOR:PALETTE:USER:MASKHighlight

This command sets or queries the user palette mask hits color. It assigns the hue-lightness-saturation (HLS) triplet to be used for the mask highlight (mask hits) color for the user palette.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:COLOR:PALETTE:USER:MASKHighlighT <NR1>,<NR1>,<NR1> DISplay:COLOR:PALETTE:USER:MASKHighlighT?
<b>Arguments</b>	<NR1> Hue. Range of 0 to 360. <NR1> Lightness. Range of 0 to 100. <NR1> Saturation. Range of 0 to 100.
<b>Examples</b>	DISPLAY:COLOR:PALETTE:USER:MASKHIGHLIGHT330,75,100 sets the mask highlight (mask hits) color for the user palette.  DISPLAY:COLOR:PALETTE:USER:MASKHIGHLIGHT? might return :DISPLAY:COLOR:PALETTE:USER:MASK:HIGHLIGHT 325,65,100, indicating that the user palette mask highlight color is assigned a hue of 325, a lightness of 65, and a saturation of 100.

## DISplay:COLOR:PALEtte:USER:MATH<x>

This command sets or queries the user palette math colors. It assigns the hue-lightness-saturation (HLS) triplet to be used for the specified math waveform for the user palette. The math waveform is specified by x. The value of x can range from 1 through 4.

**Group** Display Control

**Syntax** DISplay:COLOR:PALEtte:USER:MATH<x> <NR1>, <NR1>, <NR1>  
DISplay:COLOR:PALEtte:USER:MATH<x>?

**Arguments** <NR1> Hue. Range of 0 to 360.  
<NR1> Lightness. Range of 0 to 100.  
<NR1> Saturation. Range of 0 to 100.

**Examples** DISPLAY:COLOR:PALETTE:USER:MATH2 330,75,100 sets the color of the specified Math 2 waveform for the user palette.

DISPLAY:COLOR:PALETTE:USER:MATH2? might return  
:DISPLAY:COLOR:PALETTE:USER:MATH2 345,75,98, indicating  
that the user palette color of the Math 2 waveform is set to a hue of 345, a  
lightness of 75, and a saturation of 98.

## DISplay:COLOR:PALEtte:USER:REF<x>

This command sets or queries the user palette reference colors assigned to the reference waveforms. It assigns the hue-lightness-saturation (HLS) triplet to be used for the specified reference waveform color for the user palette. The reference waveform is specified by x. The value of x can range from 1 through 4.

**Group** Display Control

**Syntax** DISplay:COLOR:PALEtte:USER:REF<x> <NR1>, <NR1>, <NR1>  
DISplay:COLOR:PALEtte:USER:REF<x>?

**Arguments** <NR1> Hue. Range of 0 to 360.  
<NR1> Lightness. Range of 0 to 100.  
<NR1> Saturation. Range of 0 to 100.

**Examples** `DISPLAY:COLOR:PALETTE:USER:REF1 330,75,100` sets the Reference 1 waveform color for the user palette.

`DISPLAY:COLOR:PALETTE:USER:REF2?` might return `:DISPLAY:COLOR:PALETTE:USER:REF2 300,90,100`, indicating that the user palette color for Reference 2 waveform is assigned a hue of 300, and lightness of 90, and a saturation of 100.

## DISplay:COLOr:PALEtte:USEr:WAVEform

This command sets or queries the user palette waveform colors. It assigns the hue-lightness-saturation (HLS) triplet to be used for the specified waveform for the user palette.

**Group** Display Control

**Syntax** `DISplay:COLOr:PALEtte:USEr:WAVEform`  
`{HLS|TEMPERature|SPECTra1}`  
`DISplay:COLOr:PALEtte:USEr:WAVEform?`

**Arguments** `HLS` sets the color of user waveforms to the hue, lightness, and saturation specified by the [DISplay:COLOr:PALEtte:USEr:WAVEform](#) commands.

`TEMPERature` sets the color of user waveforms to temperature.

`SPECTra1` sets the color of user waveforms to spectral.

**Examples** `DISPLAY:COLOR:PALETTE:USER:WAVEFORMSPECTRAL` sets the user waveforms color palette to spectral.

`DISPLAY:COLOR:PALETTE:USER:WAVEFORM?` might return `:DISPLAY:COLOR:PALETTE:USER:WAVEFORM TEMPERATURE`, indicating that the color of user waveforms is set to temperature.

## DISplay:COLOr:REFCOLOr

This command sets or queries the color to be used for reference traces, either in the standard palette's nominal REF color or according to the color of the source waveform. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Colors tab.

**Group** Display Control

<b>Syntax</b>	DISplay:COLOR:REFCOLOR {DEFAULT INHERIT} DISplay:COLOR:REFCOLOR?
<b>Related Commands</b>	DISplay:INTENSITY:WAVEform:IMAGEView, DISplay:INTENSITY:WAVEform:RECORDView
<b>Arguments</b>	DEFAULT assigns color reference traces to the nominal palette reference color, which is off-white.  INHERIT assigns color reference traces to the source waveform color.
<b>Examples</b>	DISPLAY:COLOR:REFCOLORDEFAULT assigns color reference traces to the nominal palette reference color, which is off-white.  DISPLAY:COLOR:REFCOLOR? might return :DISPLAY:COLOR:REFCOLOR INHERIT, indicating that source waveform color is used for reference traces.

## DISplay:DIGital:HEIght

This command sets or queries the height of the digital input waveform and the label associated with the channel.

---

**NOTE.** This command is available only on the MSO70000 series of products.

---

<b>Group</b>	Digital
<b>Syntax</b>	DISplay:DIGital:HEIght {SMAl1 LARGe MEDIum XSMAl1} DISplay:DIGital:HEIght?
<b>Arguments</b>	SMAl1 specifies the height of the digital input waveform and the label associated with the channel to small.  LARGe specifies the height of the digital input waveform and the label associated with the channel to large.  MEDIum specifies the height of the digital input waveform and the label associated with the channel to medium.  XSMAl1 specifies the height of the digital input waveform and the label associated with the channel to extra small.
<b>Examples</b>	DISPLAY:DIGITAL:HEIGHTMEDIUM sets the height of the digital input waveform and the label associated with the channel to MEDIUM.

DISPLAY:DIGITAL:HEIGHT? might return DISPLAY:DIGITAL:HEIGHTSMALL, indicating that the height of the digital input waveform and the label associated with the channel is set to SMALL.

## DISplay:FILTer

This command sets or queries the type of interpolation to use for the display. Filtering only applies to normal-mode acquisition. The DISplay:FILTer command also provides selection for acquisition interpolation type. This command is equivalent to selecting Waveform Interpolation from the Display menu.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:FILTer {LINEAr SINX} DISplay:FILTer?
<b>Arguments</b>	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.
<b>Examples</b>	DISPLAY:FILTERSINX specifies sine-curve interpolation, when magnifying waveforms.  DISPLAY:FILTER? might return :DISPLAY:FILTER LINEAr, indicating that straight-line interpolation is specified for magnifying waveforms.

## DISplay:FORMat

This command sets or queries the display format. This command is equivalent to selecting Format from the Display menu.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:FORMat {YT XY XYZ} DISplay:FORMat?
<b>Arguments</b>	YT sets the display to a voltage versus time format and is the default mode.  XY argument displays one waveform against another. The source pairs that make up an XY trace are predefined and are listed in the following table. Selecting one

source causes its corresponding source to be implicitly selected, producing a single trace from the two input waveforms.

XYZ argument is available only for four-channel instruments. The argument combines channel 1 and channel 2 for X and Y coordinates and uses channel 3 to provide the intensity value for the sample. XYZ groups channels 1, 2 and 3 to form a single trace. Other channel, math, and reference waveforms are turned off.

**Table 2-43: XY Format Pairs**

X-Axis Source	Y-Axis Source
CH 1	CH 2
CH 3	CH 4
Ref 1	Ref 2
Ref 3	Ref 4

**Examples** `DISPLAY:FORMATYT` selects a voltage versus time format for the display.  
`DISPLAY:FORMAT?` might return `:DISPLAY:FORMAT XY` is for the display format, indicating that the display is in the XY mode.

## DISplay:GRAticule

This command selects or queries the type of graticule that is displayed. This command is equivalent to selecting Graticule Style from the Display menu.

**Group** Display Control

**Syntax** `DISplay:GRAticule`  
`{CROSSHair|FRAMe|FUL|GRId|IRE|NTSC|MV|PAL}`  
`DISplay:GRAticule?`

**Arguments** `CROSSHair` specifies a frame and cross hairs.  
`FRAMe` specifies a frame only.  
`FUL` specifies a frame, a grid and cross hairs.  
`GRId` specifies a frame and grid only.  
`IRE` specifies an IRE video graticule, and sets the vertical scale to 143 mV per division.  
`NTSC` specifies an NTSC video graticule (same as the IRE graticule), and sets the vertical scale to 133 mV per division.

MV specifies an mV video graticule and sets the vertical scale to 133 mV per division. This graticule is used to measure PAL standard video signals.

PAL specifies a PAL video graticule (same as the mV graticule) and sets the vertical scale to 133 mV per division. This graticule is used to measure PAL standard video signals.

**Examples**    `DISPLAY:GRATICULEFRAME` sets the graticule type to display the frame only.

`DISPLAY:GRATICULE?` might return `:DISPLAY:GRATICULE FULL`, indicating that all graticule elements are selected.

## DISplay:INTENSITY? (Query Only)

This query-only command returns the waveform saturation level and screen saver settings. This command is equivalent to selecting Display Setup from the Display menu and choosing the Appearance tab.

**Group**    Display Control

**Syntax**    `DISplay:INTENSITY?`

**Related Commands**    [DISplay:INTENSITY:WAVEform:IMAGEView](#), [DISplay:INTENSITY:WAVEform:RECORDView](#)

**Arguments**    None

**Examples**    `DISPLAY:INTENSITY?` might return  
`:DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW 75.0000;RECORDVIEW 10.0000;:DISPLAY:INTENSITY:AUTOBRIGHT 1.`

## DISplay:INTENSITY:WAVEform:IMAGEView

This command sets or queries the saturation level for image view waveforms.

**Group**    Display Control

**Syntax**    `DISplay:INTENSITY:WAVEform:IMAGEview <NR2>`  
`DISplay:INTENSITY:WAVEform:IMAGEview?`

**Related Commands**    [DISplay:INTENSITy:WAVEform:RECORDView](#)

**Arguments**    <NR2> is the waveform saturation and ranges from 10 to 100 percent.

**Examples**    DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW100 sets the image view waveform saturation to 100 percent.

DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW? might return  
:DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW 60.0000, indicating that the saturation level of the image view waveforms is currently set to 60 percent.

## DISplay:INTENSITy:WAVEform:RECORDView

This command sets or queries the saturation level for record view waveforms.

**Group**    Display Control

**Syntax**    DISplay:INTENSITy:WAVEform:RECORDview <NR2>  
DISplay:INTENSITy:WAVEform:RECORDview?

**Related Commands**    [DISplay:INTENSITy:WAVEform:IMAGEView](#)

**Arguments**    <NR2> is the waveform saturation and ranges from 10 to 100 percent.

**Examples**    DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW100 sets the record view waveform saturation to 100 percent.

DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW? might return  
:DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW 60.0000, indicating that the saturation of the record view waveforms is currently set to 60 percent.

## DISplay:PERSiStence

This command sets or queries the persistence aspect of the display. This affects the display only and is equivalent to selecting Display Persistence from the Display menu.

**Group**    Display Control



**Syntax** `DISplay:PERSiStence {OFF|INFPersist|VARpersist}`  
`DISplay:PERSiStence?`

**Related Commands** [DISplay:VARpersist](#)

**Arguments** OFF disables the persistence aspect of the display.  
 INFPersist 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.

**Examples** `DISPLAY:PERSISTENCEVARPERSIST` 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 query-only command controls the clearing of persistence data.  
 This is equivalent to pressing the front panel 'Clear' button.

**Group** Display Control

**Syntax** `DISplay:PERSiStence:RESEt`

**Related Commands** [DISplay:PERSiStence](#), [DISplay:VARpersist](#)

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

## DISplay:SCREENExt? (Query Only)

This query-only command returns all screen text settings.

**Group** Display Control

**Syntax** `DISplay:SCREENExt?`

**Related Commands** [DISplay:SCREExt:STATE](#), [DISplay:SCREExt:LABel<x>?](#), [DISplay:SCREExt:LABel<x>:NAME](#), [DISplay:SCREExt:LABel<x>:XPOS](#), [DISplay:SCREExt:LABel<x>:YPOS](#)

**Examples** DISPLAY:SCREExt? might reply: :DISPLAY:SCREExt:STATE  
 0;LABEL1:NAME "";XPOS 100;YPOS  
 5;:DISPLAY:SCREExt:LABEL2:NAME "";XPOS 100;YPOS  
 20;:DISPLAY:SCREExt:LABEL3:NAME "";XPOS 100;YPOS  
 35;:DISPLAY:SCREExt:LABEL4:NAME "";XPOS 100;YPOS  
 50;:DISPLAY:SCREExt:LABEL5:NAME "";XPOS 100;YPOS  
 65;:DISPLAY:SCREExt:LABEL6:NAME "";XPOS 100;YPOS  
 80;:DISPLAY:SCREExt:LABEL7:NAME "";XPOS 100;YPOS  
 95;:DISPLAY:SCREExt:LABEL8:NAME "";XPOS 100;YPOS 110.

## DISplay:SCREExt:LABel<x>? (Query Only)

This query-only command returns the screen text setting for a given label. Specifically, it provides the name, horizontal position (XPOS), and vertical position (YPOS).

**Group** Display Control

**Syntax** DISplay:SCREExt:LABel<x>?

**Related Commands** [DISplay:SCREExt?](#), [DISplay:SCREExt:STATE](#), [DISplay:SCREExt:LABel<x>:NAME](#), [DISplay:SCREExt:LABel<x>:XPOS](#), [DISplay:SCREExt:LABel<x>:YPOS](#)

**Examples** DISPLAY:SCREExt:LABEL3? might return  
 :DISPLAY:SCREExt:LABEL3:NAME "THIS IS SCREEN TEXT"; :XPOS  
 100; :YPOS 35, indicating that the name of Label 3 is THIS IS SCREEN  
 TEXT, and the label is position at 100 horizontally and at 35 vertically.

## DISplay:SCREExt:LABel<x>:FONTColor

This command sets or queries the screen text label font color. <x> is the label number, 1 through 8.

**Group** Display Control

**Syntax**    `DISplay:SCREENText:LABel<x>:FONTCOLOR <QString>`  
`DISplay:SCREENText:LABel<x>:FONTCOLOR?`

**Arguments**    `<QString>` argument is the font color to be displayed for a given label.

**Examples**    `DISPLAY:SCREENTEXT:LABEL1:FONTCOLOR 0240120` sets the Label 1 font color to 0240120 with Hue of 0, Lightness of 240, and Saturation of 120.

`DISPLAY:SCREENTEXT:LABEL3:FONTCOLOR?` might return  
`:DISPLAY:SCREENTEXT:LABEL3:FONTCOLOR 16711680`, indicating that the font color of the Label 1 is set to 16711680 with Hue of 167, Lightness of 116, and Saturation of 80.

## DISplay:SCREENText:LABel<x>:FONTName

This command sets or queries the screen text label font name. `<x>` is the label number which ranges from 1 through 8.

**Group**    Display Control

**Syntax**    `DISplay:SCREENText:LABel<x>:FONTName <QString>`  
`DISplay:SCREENText:LABel<x>:FONTName?`

**Arguments**    `<QString>` argument is the font name to be displayed for a given label.

**Examples**    `DISPLAY:SCREENTEXT:LABEL1:FONTNAME ARIAL` sets the Label 1 font name to Arial.

`DISPLAY:SCREENTEXT:LABEL1:FONTNAME?` might return  
`DISPLAY:SCREENTEXT:LABEL1:FONTNAME "Arial Narrow"`, indicating that the Label 1 font name is set to Arial Narrow.

## DISplay:SCREENText:LABel<x>:FONTSize

This command sets or queries the screen text label font size. `<x>` is the label number which ranges from 1 through 8.

**Group**    Display Control

**Syntax**     `DISPlay:SCREENText:LABel<x>:FONTSize <NR1>`  
`DISPlay:SCREENText:LABel<x>:FONTSize?`

**Arguments**     `<NR1>` is the font size to be displayed for a given label.

**Examples**     `DISPLAY:SCREENTEXT:LABEL1:FONTSIZE 11` sets the Label 1 font size to 11.  
`DISPLAY:SCREENTEXT:LABEL1:FONTSIZE?` might return  
`DISPLAY:SCREENTEXT:LABEL1:FONTSIZE 9`, indicating that the  
Label 1 font size is set to 9.

## DISPlay:SCREENText:LABel<x>:FONTSTyle

This command sets or queries the screen text label font style. `<x>` is the label number, 1 through 8.

**Group**     Display Control

**Syntax**     `DISPlay:SCREENText:LABel<x>:FONTSTyle <QString>`  
`DISPlay:SCREENText:LABel<x>:FONTSTyle?`

**Arguments**     `<QString>` argument is the font style to be displayed for a given label.

**Examples**     `DISPLAY:SCREENTEXT:LABEL1:FONTSTYLE BOLD` sets the Label 1 font style  
to BOLD.  
`DISPLAY:SCREENTEXT:LABEL1:FONTSTYLE?` might return  
`DISPLAY:SCREENTEXT:LABEL1:FONTSTYLE "Regular"`, indicating that the  
Label 1 font style is Regular.

## DISPlay:SCREENText:LABel<x>:NAME

This command sets the text to be displayed for a given label. The label is specified by `<x>`. The value of `<x>` can range from 1 through 8.

**Group**     Display Control

**Syntax**     `DISPlay:SCREENText:LABel<x>:NAME <QString>`  
`DISPlay:SCREENText:LABel<x>:NAME?`

<b>Related Commands</b>	, <a href="#">DISplay:SCREENText:STATE</a> , <a href="#">DISplay:SCREENText:LABel&lt;x&gt;:XPOS</a> , <a href="#">DISplay:SCREENText:LABel&lt;x&gt;:YPOS</a>
<b>Arguments</b>	<QString> argument is the text to be displayed for a given label.
<b>Examples</b>	<p>DISPLAY:SCREENTEXT:LABEL2:NAME "THIS IS SCREEN TEXT" sets the screen text for Label 2 to THIS IS SCREEN TEXT.</p> <p>DISplay:SCREENText:LABEL2:NAME? might return :DISPLAY:SCREENTEXT:LABEL2:NAME "Mywaveform", indicating that the Label 2 screen text is set to MyWaveform.</p>

## DISplay:SCREENText:LABel<x>:STATE

This command sets the state to be displayed for a given label. <x> is the label number, 1 through 8.

<b>Group</b>	Display Control
<b>Syntax</b>	<pre>DISplay:SCREENText:LABel&lt;x&gt;:STATE {&lt;NR1&gt; OFF ON} DISplay:SCREENText:LABel&lt;x&gt;:STATE?</pre>
<b>Arguments</b>	<p>OFF sets the screen text to not display.</p> <p>ON sets the screen text to display.</p>
<b>Examples</b>	<p>DISPLAY:SCREENTEXT:LABEL2:NAME ON sets the Label 2 screen text to display.</p> <p>DISplay:SCREENText:LABEL2:STATE? might return :DISPLAY:SCREENTEXT:LABEL2:STATE 1, indicating that the Label 2 screen text display is ON.</p>

## DISplay:SCREENText:LABel<x>:XPOS

This command sets the horizontal position of a given screen text label. The label is specified by x. The value of x can range from 1 through 8.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:SCREENText:LABel<x>:XPOS <NR1>

<b>Related Commands</b>	<a href="#">DISplay:SCREExt:STATE</a> , <a href="#">DISplay:SCREExt:LABel&lt;x&gt;:NAME</a> , <a href="#">DISplay:SCREExt:LABel&lt;x&gt;:YPOS</a>
<b>Arguments</b>	<NR1> sets the horizontal position of the screen text label. The value can range from 0 to 500.
<b>Examples</b>	<p>DISPLAY:SCREExt:LABel3:XPOS 100 sets the horizontal position of Label 3 to 100.</p> <p>DISPLAY:SCREExt:LABel3:XPOS? might return :DISPLAY:SCREExt:LABel3:XPOS 250, indicating that the screen Label 3 is horizontally positioned to 250.</p>

## DISplay:SCREExt:LABel<x>:YPOS

This command sets the vertical position of a given label. The label is specified by x. The value of x can range from 1 through 8.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:SCREExt:LABel<x>:YPOS <NR1>
<b>Related Commands</b>	<a href="#">DISplay:SCREExt:STATE</a> , <a href="#">DISplay:SCREExt:LABel&lt;x&gt;:NAME</a> , <a href="#">DISplay:SCREExt:LABel&lt;x&gt;:XPOS</a>
<b>Arguments</b>	<NR1> sets the vertical position of the screen text label. The value can range from 0 to 385.
<b>Examples</b>	<p>DISPLAY:SCREExt:LABel3:YPOS 100 sets the vertical position of Label 3 to 100.</p> <p>DISPLAY:SCREExt:LABel3:YPOS? might return :DISPLAY:SCREExt:LABel3:YPOS 192, indicating that screen Label 3 is vertically positioned to 192.</p>

## DISplay:SCREExt:STATE

This command controls the display of screen text.

<b>Group</b>	Display Control
--------------	-----------------

**Syntax** `DISPlay:SCREENText:STATE {ON|OFF|<NR1>}`

**Related Commands** [DISplay:SCREENText?](#), [DISplay:SCREENText:LABel<x>?](#), [DISplay:SCREENText:LABel<x>:NAME](#), [DISplay:SCREENText:LABel<x>:XPOS](#), [DISplay:SCREENText:LABel<x>:YPOS](#)

**Arguments** ON turns on the display of screen text.  
OFF turns off the display of screen text.  
<NR1> = 0 disables screen text; any other value enables screen text.

**Examples** `DISPLAY:SCREENTEXT:STATEON` turns the screen text display on.  
`DISPLAY:SCREENTEXT:STATE?` might return `:DISPLAY:SCREENTEXT:STATE 1`, indicating that the screen text is displayed.

## DISplay:SHOWREmote

This command sets or queries the state of the remote display feature and is equivalent to selecting Display Remote from the Display menu. The query form of this command returns ON (1) or OFF (0). This command is not available for MSO70000 Series instruments.

This feature allows you to view waveforms and other graticule data on a remote display using remote control software like VNC (Virtual Network Computing) or Symantec pcAnywhere.

**Group** Display Control

**Syntax** `DISPlay:SHOWREmote {ON|OFF|NR1}`  
`DISPlay:SHOWREmote?`

**Arguments** ON enables the remote display of waveform and other graticule data.  
OFF disables the remote display of waveform and other graticule data.  
NR1 = 0 disables remote display of waveform and other graticule data; any other value enables remote display of waveform and other graticule data.

**Examples** `DISPLAY:SHOWREMOTEON` enables remote display of waveform and other graticule data.

DISPLAY:SHOWREMOTE? might return :DISPLAY:SHOWREMOTE 1, indicating that the remote display of waveform and other graticule data is currently enabled.

## DISplay:STYLE

This command sets or queries how the data is displayed for normal and FastAcq modes. This command is equivalent to selecting Display Style from the Display menu and choosing a style.

**Group** Display Control

**Syntax** DISplay:STYLE {VEctors|DOTs|INTENSIFied}  
DISplay:STYLE?

**Arguments** VEctors connects adjacent data points. New points immediately replace old ones.  
DOTs displays individual data points. New points immediately replace old ones.  
INTENSIFied causes the display to show interpolated samples with dark spaces (Only the “real” samples are displayed).  
When FastAcq mode is enabled, intensified samples display as dots. However, turning off FastAcq mode causes the display style to snap back to its previous setting.

**Examples** DISPLAY:STYLE VECTORS sets the display to connect adjacent data points.  
DISPLAY:STYLE? might return :DISPLAY:STYLE INTENSIFIED, indicating that interpolated samples are not displayed.

## DISplay:TRIGBar

This command controls or queries the display of the trigger-level indicator bars. Indicator bars show where the trigger voltage level is set.

The instrument will only display the bar if the associated trigger source is also displayed. If both a main and a delayed trigger are displayed, then two bars will appear. One will accompany each source. If a logic trigger is selected, then multiple bars may appear. One will show the upper threshold and one will show the lower threshold. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Objects tab.

**Group** Display Control



<b>Syntax</b>	<code>DISPlay:TRIGBar {OFF SHORT LONG}</code> <code>DISPlay:TRIGBar?</code>
<b>Arguments</b>	<p><code>OFF</code> removes the trigger indicator bar from the display.</p> <p><code>SHORT</code> displays, as the indicator, a short arrow at the right side of the graticule for each displayed trigger signal.</p> <p><code>LONG</code> displays, as the indicator, a horizontal line across the width of the graticule for each displayed trigger signal.</p>
<b>Examples</b>	<p><code>DISPLAY:TRIGBARLONG</code> sets the display to show a long trigger indicator bar (or bars).</p> <p><code>DISPLAY:TRIGBAR?</code> might return <code>:DISPLAY:TRIGBAR OFF</code>, indicating that the indicator bar is removed from the display.</p>

## DISPlay:TRIGT

This command controls or queries the display of the trigger T. The trigger T shows where the trigger occurred on the waveform.

<b>Group</b>	Display Control
<b>Syntax</b>	<code>DISPlay:TRIGT {&lt;NR1&gt; OFF ON}</code> <code>DISPlay:TRIGT?</code>
<b>Arguments</b>	<p><code>&lt;NR1&gt; = 0</code> disables the trigger T; any other value displays the trigger T.</p> <p><code>OFF</code> removes the trigger indicator T from the display.</p> <p><code>ON</code> displays a T at the trigger point.</p>
<b>Examples</b>	<p><code>DISPLAY:TRIGT OFF</code> disables the trigger T.</p> <p><code>DISPLAY:TRIGT?</code> might return <code>:TRIGT 1</code> indicating that the trigger T is displayed.</p>

## DISPlay:VARpersist

This command sets or queries persistence decay time, which is the approximate decay time for a freshly struck persistence sample. This command is equivalent to selecting Display Setup from the Display menu, selecting the Appearance tab, and then entering the desired Persist Time.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:VARpersist <NR3> DISplay:VARpersist?
<b>Related Commands</b>	<a href="#">DISplay:PERsistence</a>
<b>Arguments</b>	<NR3> indicates the persistence decay time and ranges from 0.5 to 100.
<b>Examples</b>	DISPLAY:VARPERSIST5 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 controls or queries the display of acquisition and reference waveforms.

<b>Group</b>	Display Control
<b>Syntax</b>	DISplay:WAVEform {<NR1> OFF ON} DISplay:WAVEform?
<b>Arguments</b>	<NR1> = 0 removes waveforms from the display; any other value displays the waveforms. OFF removes the waveforms from the display. ON displays waveforms on the display.
<b>Examples</b>	DISPLAY:WAVEFORMOFF removes waveforms from the display. DISPLAY:WAVEFORM? might return :WAVEFORM 1, indicating that waveforms are displayed.

## D<x>:LABEL

This command sets or queries the label that appears for the specified digital input on the display.

**NOTE.** *This command is available only on the MSO70000 series of products.*

<b>Group</b>	Digital
<b>Syntax</b>	D<x>:LABEL <QString> D<x>:LABEL?
<b>Related Commands</b>	D<x>:POSition, D<x>:THRESHold
<b>Arguments</b>	<QString> is an alphanumeric string of characters, enclosed in quotes, that defines the label text.
<b>Examples</b>	D1:LABEL "INPUT" sets the label for the digital input D1 to INPUT. D1:LABEL? might return :D1:LABEL "CLOCK", indicating that the label for the digital input D1 is set to CLOCK.

## D<x>:POSition

This command sets or queries the position for the specified digital input.

**NOTE.** *This command is available only on the MSO70000 series of products.*

<b>Group</b>	Digital
<b>Syntax</b>	D<x>:POSition <NR3> D<x>:POSition?
<b>Related Commands</b>	D<x>:LABEL, D<x>:THRESHold CH<x>:POSition, D<x>:POSition, MATH<x>:VERTical:POSition
<b>Arguments</b>	<NR3> specifies the position.
<b>Examples</b>	D1:POSITION "INPUT" sets the position for the digital input D1 to 2 units. D1:POSITION? might return :D1:POSITION CLOCK, indicating that the position for the digital input D1 is set to 3 units.

## D<x>:PROBE:ID:SERnumber? (Query Only)

This command queries the serial number of the digital probe that provides the specified digital signal.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Digital

**Syntax** D<x>:PROBE:ID:SERnumber?

**Related Commands** [D<x>:PROBE:ID:TYPE?](#)

**Examples** D1:PROBE:ID:SERNUMBER? might return :D1:PROBE:ID:SERNUMBER "PQ009", indicating that the serial number of the digital probe that provides the specified digital signal D1 is set to PQ009.

## D<x>:PROBE:ID:TYPE? (Query Only)

This command queries the type of digital probe that provides the specified digital signal.

---

**NOTE.** *This command is available only on the MSO70000 series of products.*

---

**Group** Digital

**Syntax** D<x>:PROBE:ID:TYPE?

**Related Commands** [D<x>:PROBE:ID:SERnumber?](#)

**Examples** D1:PROBE:ID:TYPE? might return :D1:PROBE:ID:TYPE "P6717", indicating that the type of digital probe which provides the specified digital signal D1 is P6717.

## D<x>:THRESHold

This command sets or queries the threshold level for the specified digital signal.

**NOTE.** *This command is available only on the MSO70000 series of products.*

<b>Group</b>	Digital
<b>Syntax</b>	D<x>:THRESHold <NR3> D<x>:THRESHold?
<b>Related Commands</b>	D<x>:LABEL, D<x>:POSition CH<x>:THRESHold, MATH<x>:THRESHold
<b>Arguments</b>	<NR3> specifies the threshold level, in volts.
<b>Examples</b>	D5:THRESHOLD 1.5 sets the threshold for digital channel D5 to 1.5 volts. D14:THRESHOLD might return :D14:THRESHOLD 1.2500 indicating that the threshold level for the digital channel D14 is set to 1.25 volts.

## E-Mail (No Query Form)

This command (no query form) sends a test e-mail message or sets the current e-mail sent count to zero.

<b>Group</b>	E-mail
<b>Syntax</b>	EMail {TEST RESET}
<b>Arguments</b>	TEST argument sends a test e-mail message. RESET argument sets the e-mail sent count to zero.
<b>Examples</b>	EMAILTEST This command sends a test e-mail. EMAILRESET resets the sent mail count to zero.

## E-Mail:ATtempts

This command sets or queries the number of times that an attempt will be made to send e-mail to the SMTP e-mail server. The default is 1.

<b>Group</b>	E-mail
<b>Syntax</b>	EMail:ATTEMPTS <NR1> EMail:ATTEMPTS?
<b>Arguments</b>	<NR1> can be 1 through 20. This number is the specified number of attempts that will be made to send e-mail to the SMTP server.
<b>Examples</b>	EMAIL:ATTEMPTS15 specifies that 15 attempts will be made to send the e-mail to the SMTP e-mail server.  EMAIL:ATTEMPTS? might return :EMAIL:ATTEMPTS 10, indicating that 10 attempts will be made to send the e-mail to the SMTP e-mail server.

## Email:AUTHLogin

This command sets or queries the login name that will be used if the SMTP e-mail server requires one for authentication.

<b>Group</b>	E-mail
<b>Syntax</b>	EMail:AUTHLogin {<QString>} EMail:AUTHLogin?
<b>Arguments</b>	<QString> argument is a string that specifies the login name to be used with the SMTP e-mail server.
<b>Examples</b>	EMAIL:AUTHLOGINKentb specifies that kentb will be used as the login name with the SMTP e-mail server.  EMAIL:AUTHLOGIN? might return :EMAIL:AUTHLOGIN "bobk", indicating that bobk will be used as the login name used with the SMTP e-mail server.

## Email:AUTHPassword (No Query Form)

This command (no query form) sets the password that will be used if the SMTP e-mail server requires one for authentication.

<b>Group</b>	E-mail
--------------	--------

<b>Syntax</b>	EMail:AUTHPassword {<QString>}
<b>Arguments</b>	<QString> argument is a string that specifies the password to be used with the SMTP e-mail server.
<b>Examples</b>	EMAIL:AUTHPASSWORD "123xyz" specifies that 123xyz will be used as the password with the SMTP e-mail server.

## EMail:COUNT? (Query Only)

This query only command returns the number of e-mails that have been sent since Email on Event was armed. The range of values returned can be from 0 to 50.

<b>Group</b>	E-mail
<b>Syntax</b>	EMail:COUNT?
<b>Examples</b>	EMAIL:COUNT? might return :EMAIL:COUNT 15, indicating that 15 e-mails have been sent since e-mail on event was armed.

## EMail:FROM

This command sets or queries the From line in the e-mail. The default for the From line is the instrument model and serial number.

<b>Group</b>	E-mail
<b>Syntax</b>	EMail:FROM {<QString>} EMail:FROM?
<b>Arguments</b>	<QString> argument is a string that is placed in the From line of the e-mail. An example is "johnz".
<b>Examples</b>	EMAIL:FROM"kentb" specifies that kentb will be placed in the From line of the e-mail.  EMAIL:FROM? might return :EMAIL:FROM "BOBK", indicating that bobk will appear in the From line of the e-mail.

## EMail:HOSTwanted

This command sets or queries the host name that will be used when e-mail is sent to the SMTP e-mail server if the DPO host name will not work.

**Group** E-mail

**Syntax** EMail:HOSTwanted {<QString>}  
EMail:HOSTwanted?

**Arguments** <QString> argument is a string that specifies the host name to be used with the SMTP e-mail server if the default DPO host name will not work.

**Examples** EMAIL:HOSTWANTED"LazyEight" specifies that LazyEight will be used as the host name when sending e-mail to the SMTP e-mail server.

EMAIL:HOSTWANTED? might return :EMAIL:HOSTWANTED "LazyEight", indicating that LazyEight is the host name used with the SMTP e-mail server.

## EMail:IMAGe

This command sets or queries whether a screen image is included in e-mail. If this command is set to ON, the format and content of the screen image included is specified using the EXPORT commands within the Hard Copy Command group.

**Group** E-mail

**Syntax** EMail:IMAGe {ON|OFF|NR1}  
EMail:IMAGe?

**Related Commands** [EXPORT](#)

**Arguments** ON argument enables the inclusion of a screen image in e-mail.

OFF argument disables the inclusion of a screen image in e-mail.

<NR1> = 0 disables the inclusion of a screen image in e-mail; any other value enables the inclusion of a screen image in e-mail.



**Examples** EMAIL:IMAGE1 enables the inclusion of a screen image in e-mail.

EMAIL:IMAGE? might return :EMAIL:IMAGE 0 indicating that a screen image is not included in the e-mail.

## EMail:MASK

This command sets or queries whether e-mail is sent when a mask test failure occurs.

**Group** E-mail

**Syntax** EMail:MASK {ON|OFF|<NR1>}  
EMail:MASK?

**Arguments** ON argument enables sending e-mail when a mask test failure occurs.

OFF disables sending e-mail when a mask test failure occurs.

<NR1> = 0 disables sending e-mail when a mask test failure occurs; any other value enables sending e-mail when a mask test failure occurs.

**Examples** EMAIL:MASK1 enables sending e-mail when a mask test failure occurs.

EMAIL:MASK? might return :EMAIL:MASK 0, indicating that e-mail will not be sent when a mask test failure occurs.

## EMail:MAXSize

This command sets or queries the maximum size (in megabytes) of e-mail that can be sent to the SMTP server.

**Group** E-mail

**Syntax** EMail:MAXSize {<NR1>}  
EMail:MAXSize?

**Arguments** <NR1> can be 0 through 2000 (megabytes). This is the maximum size of each e-mail that can be sent to the SMTP server.

**Examples** EMAIL:MAXSIZE5 sets the maximum size of each e-mail enclosure to 5 megabytes of data.

EMAIL:MAXSIZE? might return :EMAIL:MAXSIZE 10, indicating that each e-mail sent to the SMTP server can contain up to 10 megabytes of data.

## EMail:MEASUrement

This command sets or queries whether measurement data is included as an attachment to e-mail. You must specify the saved measurement selection and the measurement format using the Email on Event setup menu. There are no remote commands for this purpose.

**Group** E-mail

**Syntax** EMail:MEASUrement {ON|OFF|<NR1>}  
EMail:MEASUrement?

**Arguments** ON enables the inclusion of measurement data in e-mail.  
OFF disables the inclusion of measurement data in e-mail.  
<NR1> = 0 disables the inclusion of measurement data in the e-mail; any other value enables including measurement data in the e-mail.

**Examples** EMAIL:MEASUREMENT1 enables the inclusion of measurement data in e-mail.  
EMAIL:MEASUREMENT? might return :EMAIL:MEASUREMENT 0, indicating that measurement data is not included in the e-mail.

## EMail:NUMEMails

This command sets or queries the number of e-mails you can send when Email on Event is armed, from 1 to 50. This limit is to restrict e-mail floods.

**Group** E-mail

**Syntax** EMail:NUMEMails {<NR1>}  
EMail:NUMEMails?

**Related Commands** [EMail](#)

**Arguments** <NR1> can be 1 through 50. This is the number of e-mails you can send before the number of sent e-mails must be reset with the EMail RESet command.

**Examples** EMAIL:NUMEMAILS10 sets the number of e-mails that can be sent to 10.

EMAIL:NUMEMAILS? might return :EMAIL:NUMEMAILS 5, indicating that five e-mails can be sent.

## EMail:SMTPPort

This command sets or queries the SMTP port number that the e-mail server uses if other than the default of 25.

**Group** E-mail

**Syntax** EMail:SMTPPort {<NR1>}  
EMail:SMTPPort?

**Arguments** <NR1> can be 1 through 65535. This number specifies the TCPIP port number.

**Examples** EMAIL:SMTPPORT55 specifies that the SMTP e-mail server will communicate on TCPIP port 55.

EMAIL:SMTPPORT? might return :EMAIL:SMTPPORT 10, indicating that TCPIP port 10 will be used to communicate with the SMTP e-mail server.

## EMail:SMTPServer

This command sets or queries the address of the SMTP mail server.

**Group** E-mail

**Syntax** EMail:SMTPServer <QString>  
EMail:SMTPServer?

**Arguments** <QString> argument is the address of the SMTP mail server that will handle the mail service request. For example, "smtp.tek.com".

**Examples** EMAIL:SMTPSERVER"smtp.tek.com" sets the SMTP mail server to smtp.tek.com. This will be the mail server used for mail service requests.

EMAIL:SMTPSERVER? might return :EMAIL:SMTPSERVER "SMTP.TEK.COM", indicating that the SMTP mail sever is set to SMTP.tek.com.

## EMail:STATUS? (Query Only)

This query only command returns the status of the last e-mail you attempted to send.

**Group** E-mail

**Syntax** EMail:STATUS?

**Returns** The status might be one of the following strings:

- PASS - indicates that the last e-mail message sent was successful or that no email message was attempted.
- FAIL - indicates that the last e-mail message attempted was not sent.
- SEND - indicates that an e-mail message is currently being sent.
- HOST\_FAIL - indicates that the instrument failed to connect to the SMTP server.
- CONNECTION\_FAIL - indicates that the SMTP server refused the connection.

**Examples** EMAIL:STATUS? might return :EMAIL:STATUS FAIL, indicating that the e-mail failed to be sent.

## EMail:TIMEOut

This command sets or queries the global timeout in seconds. The default is 30 seconds. You use this timeout for socket connections and may need to change it from the default on some networks.

**Group** E-mail

**Syntax** EMail:TIMEOut <NR1>  
EMail:TIMEOut?

**Arguments** <NR1> argument is the global timeout in seconds. The value for NR1 can be 0 through 500.

**Examples** EMAIL:TIMEOUT60 that 60 seconds will be used for network connection timeouts.

EMAIL:TIMEOUT? might return :EMAIL:TIMEOUT 30, indicating that the default of 30 seconds is set for network connection timeouts.

## E-Mail:TO

This command sets or queries the address of the recipient(s) of an e-mail.

**Group** E-mail

**Syntax** EMAIL:TO <QString>  
EMAIL:TO?

**Arguments** <QString> argument is the e-mail address of the recipient (or recipients). Multiple addresses are separated with semicolons (;). For example, "johndoe@tek.com;billsmith@tek.com" specifies that both johndoe and billsmith at tek.com will receive e-mail when the event occurs. The length of this string is limited to 252 characters.

**Examples** EMAIL:TO"johndoe@tek.com" will send e-mail to johndoe@tek.com when the event occurs.

EMAIL:TO? might return :EMAIL:TO "billsmith@tek.com", indicating that the billsmith@tek.com will receive e-mail when the event occurs.

## E-Mail:TRIGger

This command sets or queries whether e-mail is sent when a trigger occurs. The e-mail is sent after the acquisition is complete.

**Group** E-mail

**Syntax** EMAIL:TRIGger {ON|OFF|<NR1>}  
EMAIL:TRIGger?

**Arguments** ON argument enables sending e-mail when a trigger occurs.

OFF disables sending e-mail when a trigger occurs.

<NR1> = 0 disables sending e-mail when a trigger occurs; any other value enables sending e-mail when a trigger occurs.

**Examples** EMAIL:TRIGGER1 enables sending e-mail when a trigger occurs.

EMAIL:TRIGGER? might return :EMAIL:TRIGGER 0, indicating that e-mail will not be sent when a trigger occurs.

## EMail:WAVEform

This command sets or queries whether waveform data is included in e-mail. The waveforms to be included, the start and stop points (and, or, frames for Fast Frame mode), and the data formats are specified by the [DATA](#) and [WFMOutpre?](#) commands within the Waveform Transfer Command group.

**Group** E-mail

**Syntax** EMail:WAVEform {ON|OFF|<NR1>}  
EMail:WAVEform?

**Arguments** ON enables the inclusion of waveform data in e-mail.

OFF disables the inclusion of waveform data in e-mail.

<NR1> = 0 disables the inclusion of waveform data in the e-mail; any other value enables including waveform data in the e-mail.

**Examples** EMAIL:WAVEFORM1 enables the inclusion of waveform data in e-mail.

EMAIL:WAVEFORM? might return :EMAIL:WAVEFORM 0, indicating that waveform data is not included in the e-mail.

## \*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** \*ESE209 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 213, showing that the SESR contains the binary value 11010101.

## 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 event codes that are in the Event 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.

## EXPort

This command sends a copy of the waveform to the file path specified by [EXPort:FILENAME](#). The EXPort query returns image format and file information.

**Group** Hard Copy

**Syntax** EXPort START  
EXPort?

**Arguments** START initiates the export.

**Examples** EXPORTSTART initiates export to the file specified file by EXPort:FILENAME.  
EXPORT? returns the export formatting and file path. The query might return :EXPORT:FILENAME "untitled"; FORMAT BMP;IMAGE NORMAL; PALETTE COLOR;VIEW FULLSCREEN "C:\TekScope\Screen Captures\hcFILE.BMP".

## EXPort:FILENAME

This command sets or queries the file/path that will be sent export data on the next [EXPort](#) command.

**Group** Hard Copy

**Syntax** EXPort:FILENAME <file path>  
EXPort:FILENAME?

**Related Commands** [HARDCopy](#)

**Arguments** <file path> specifies that the hard copy is sent to the named file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>.<drive> and one or more <dir>s are optional. The file path cannot exceed 128 characters. If you don't specify the path with <drive>: and one or more <dir>s, the default location is "C:\TekScope\Screen Captures". While filename extensions are not required, they are highly recommended.

**Examples** EXPORT:FILENAME"TEK.BMP" sets TEK.BMP as the selected file name. The next export will be sent to this file in the default hard copy directory.

EXPORT:FILENAME? might return :EXPORT:FILENAME  
"C:\TekScope\Screen Captures\TEK.BMP" as the selected file.

## EXPoRT:FORMat

This command sets or queries the image format for exporting waveforms to a file.

**Group** Hard Copy

**Syntax** EXPoRT:FORMat {BMP|JPEG|PNG|PCX|TIFF}  
EXPoRT:FORMat?

**Arguments** BMP specifies BMP image format.  
JPEG specifies JPEG image format.  
PNG specifies PNG image format.  
PCX specifies PCX image format.  
TIFF specifies TIFF image format.

**Examples** EXPORT:FORMATBMP sets the export format to BMP.  
EXPORT:FORMAT? might return :EXPORT:FORMAT JPEG, indicating that waveforms will be exported in the JPEG format.

## EXPoRT:PALEtte

This command sets or queries the export color palette.

**Group** Hard Copy

**Syntax**    `EXPoRT:PALEtTe {COLOr|INKSaver|BLACKANDWhite}`  
`EXPoRT:PALEtTe?`

**Arguments**    `COLOr` Hardcopy output is color.  
`INKSaver` Hardcopy output saves ink.  
`BLACKANDWhite` Hardcopy output is black and white.

**Examples**    `EXPORT:PALETTE` sets the export color palette to Color.  
`EXPORT:PALETTE?` might return `:EXPORT PALETTE COLOR` indicating the hard copy output will be in color.

## EXPoRT:READOUs

This command sets or queries the area on the screen where the readout appear for export. This command is equivalent to selecting Export Setup from the File menu and setting the Readouts Below Graticule in the control window.

**Group**    Hard Copy

**Syntax**    `EXPoRT:READOUs {OFFGRAticUle|ONGRAticUle}`  
`EXPoRT:READOUs?`

**Arguments**    `OFFGRATICULE` places the readouts off the graticule area.  
`ONGRATICULE` places the readouts on the graticule area for export.

**Examples**    `EXPORT:READOUTSOFFGRATICULE` places the readouts off the graticule area.  
`EXPORT:READOUTS?` might return `:EXPORT READOUTS ONGRATICULE`, indicating the readout on the graticule area for export.

## EXPoRT:VIEW

This command sets or queries the area of the screen to be exported.

**Group**    Hard Copy

**Syntax**    `EXPoRT:VIEW {FULLSCREEN|GRAticUle|FULLNOmenu}`  
`EXPoRT:VIEW?`

**Arguments**    **FULLSCREEN** displays both the graticule and menu areas of the screen.  
**GRATICULE** displays only the graticule area of the screen.  
**FULLNOMENU** displays the full screen but hides any menus or toolbars.

**Examples**    **EXPORT:VIEWFULLSCREEN** sets the exported view area to Full Screen.  
**EXPORT:VIEW?** might return **:EXPORT:VIEW GRATICULE**, indicating that only the graticule area of the screen will be exported.

## 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 Recall Default Setup from the File menu.

This command resets the instrument settings to factory defaults:

- Clears the Event Status Enable Register.
- Clears the Service Request Enable Register.
- Sets the Device Event Status Enable Register to 255.
- Sets the Power On Status Clear Flag to TRUE.
- Purges all defined aliases.
- Sets the macro defined by \*DDT to a “zero-length field”.
- Clears the pending operation flag and associated operations.

This command does not reset the following:

- The state of the GPIB (IEEE Std 488.1-1987) interface.
- The selected GPIB address.
- Calibration data that affects device specifications.
- Protected user data.
- Stored settings.
- The current password (if you set one).

**Group**    Save and Recall

**Syntax**    **FACTory**

**Related Commands**    \*PSC, \*RCL, RECALL:SETUp, \*RST, \*SAV, SAVe:SETUp

**Arguments**    None

**Examples**    FACTORY resets the instrument to its factory default settings.

## FASTAcq? (Query Only)

This query-only command returns the state of Fast Acquisitions. This command is equivalent to pressing the FASTACQ button on the front panel.

**Group**    Acquisition

**Syntax**    FASTAcq?

**Examples**    FASTACQ? might return the following string for the current acquisition parameters:  
:FASTACQ:STATE 0.

## FASTAcq:HIACQRATE

This command sets or queries the state of FastAcq optimization for capturing signal details with a higher sample rate.

**Group**    Acquisition

**Syntax**    FASTAcq:HIACQRATE {<NR1> | OFF | ON}  
FASTAcq:HIACQRATE?

**Arguments**    <NR1> = 0 sets FastAcq optimization for capturing signal details with a higher sample rate; any other value sets FastAcq optimization for capturing rare events with more acquisitions.

OFF sets FastAcq optimization for capturing signal details with a higher sample rate.

ON sets FastAcq optimization for capturing rare events with more acquisitions.

**Examples**    FASTACQ:HIACQRATE ON sets optimizes FastAcq for capturing rare events with more acquisitions.

FASTACQ:HIACQRATE? might return :FASTACQ:HIACQRATE 0, indicating that the FastAcq is optimized for capturing signal details with a higher sample rate.

## FASTAcq:STATE

This command sets or queries the state of Fast Acquisitions. This command is equivalent to the FASTACQ button on the front panel.

**Group** Acquisition

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

**Arguments** ON enables Fast Acquisitions mode.  
OFF disables Fast Acquisitions mode.  
<NR1> = 0 disables Fast Acquisitions mode; any other value enables Fast Acquisitions mode.

**Examples** FASTACQ:STATEON enables the Fast Acquisitions mode.  
FASTACQ:STATE? might return :FASTACQ:STATE 1, indicating that Fast Acquisitions mode is currently active.

## 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:PRInt](#), [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 may 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. 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"C:\MYDIR\TEK00001.SET",`  
`"C:\ANOTHERDIR\COPYOFTEK001.SET"` copies the file named  
TEK00001.SET, located in the MYDIR directory on the C drive to a file named  
COPYOFTEK001.SET in the ANOTHERDIR directory on the C drive.

## FILESystem:CWD

This command sets or queries the current working directory for FILESystem GPIB commands. The default working directory is "C:\TekScope". Anytime that you use this command to change the directory, the directory that you specify is retained as the current working directory until you either change the directory or you delete the directory. If you delete the current working directory, the instrument resets current working directory to the default directory (C:\TekScope) the next time the instrument is powered on or the next time you execute a file system command. The current working directory is retained between power cycles.

This command supports the permutations of file and directory names that are supported by Microsoft Windows:

- Relative path names; for example, “.\Temp”
- Absolute path names; for example, “C:\Windows\System”
- Implied relative path names; for example “newfile.text” becomes “C:\TekScope\newfile.txt” if the current working directory is “C:\TekScope”

**Group** File System

**Syntax** FILESystem:CWD {<new working directory path>}

**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"C:\TekScope\Screen Captures" sets the current working directory to Screen Captures.

FILESYSTEM:CWD? might return :FILESYSTEM:CWD  
"C:\TekScope\waveforms", indicating that the current working directory is set to Waveforms.

## FILESystem:DELEte (No Query Form)

This command (no query form) deletes a named file. It does not delete directories. Use the [FILESystem:RMDir](#) command to delete a directory.

**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 current working directory.



## 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 current working directory.

**Group** File System

**Syntax** FILESystem:DIR?

**Related Commands** [FILESystem:CWD](#), [FILESystem:MKDir](#)

**Arguments** None

**Examples** FILESYSTEM:DIR? returns a list of files and directories in the current working directory.

## 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 "C:\NewDirectory" creates the directory named *NewDirectory* at the root of the C drive.

These two commands create the directory MyNewSubDirectory within the existing directory MyDirectory at the root of the C drive:

```
FILESYSTEM:CWD "C:\MyDirectory"; FILESYSTEM:MKDIR  
"MyNewSubDirectory"
```

This, of course, assumes that *C:\MyDirectory* already existed and was not a read-only directory.

## FILESystem:PRInt (No Query Form)

This command (no query form) prints a named file to a named port. This command is the same as the [FILESystem:READFile](#) command.

This command is not IEEE 488.2 compliant for the following reasons:

- The command produces output on the 488.2 bus. Queries, not commands, are the only message units that should produce output on the 488.2 bus. Errors may not be recognized if they occur while data is being output in response to this command. This will be evident if a command is sent to the instrument before all of the data generated by this command has been read. In such cases, you should send a device clear message to the instrument.
- The command does not use a standard 488.2 output format. The closest defined format is the indefinite format of ARBITRARY BLOCK PROGRAM DATA encoding. However, this command output omits the initial '#0' characters needed for that format.
- Newlines, ASCII value 10, can be included in the output data. Output on the bus can be terminated by newlines. The output of this command is only terminated by <EOI>.

**Group** File System

**Syntax** FILESystem:PRInt <filepath>, GPIb

**Related Commands** [FILESystem:CWD](#), [FILESystem:READFile](#)

**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.

GPIb specifies that the hard copy is sent out of the GPIB port. GPIb is the only port allowed for this command.

**Examples** FILESYSTEM:PRINT "C:\TEK00000.IBM", GPIB sends the file named TEK00000.IBM out of the GPIB port.

## FILESystem:READFile (No Query Form)

This command (no query form) prints a named file to a named port. It is identical to the [FILESystem:PRInt](#) command.

This command is not IEEE 488.2 compliant for the following reasons:

- The command produces output on the 488.2 bus. Queries, not commands, are the only message units that should produce output on the 488.2 bus. Errors may not be recognized if they occur while data is being output in response to this command. This will be evident if a command is sent to the instrument before all of the data generated by this command has been read. In such cases, you should send a device clear message to the instrument.
- The command does not use a standard 488.2 output format. The closest defined format is the indefinite format of ARBITRARY BLOCK PROGRAM DATA encoding. However, this command output omits the initial '#0' characters needed for that format.
- Newlines, ASCII value 10, can be included in the output data. Output on the bus can be terminated by newlines. The output of this command is only terminated by <EOI>.

**Group** File System

**Syntax** FILESystem:READFile <filepath>

**Related Commands** [FILESystem:CWD](#), [FILESystem:PRInt](#)

**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:READFILE "C:\TEK00000.IBM" sends the file named TEK00000.IBM out of the GPIB port.

## FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file.

**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 name and path. If the file path is within the current working directory, you need only specify the file name.

<new 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:RENAME "C:\TEK00000.SET", "C:\MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the C drive.

## FILESystem:RMDir (No Query Form)

This command (no query form) deletes a named directory. The directory cannot contain any files. If there are files in the directory, they must first be deleted. The directory must not be a read-only directory.

**Group** File System

**Syntax** FILESystem:RMDir <directory path>

**Related Commands** [FILESystem:CWD](#)

**Arguments** <directory 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:RMDIR "C:\OldDirectory" removes the directory named OldDirectory from the root of the C drive.

## FILESystem:WRITEFile (No Query Form)

This command (no query form) copies the block data from the GPIB port to a named file.

**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> can be either DEFINITE LENGTH or INDEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA encoding as described in IEEE 488.2.

**Examples** FILESYSTEM:WRITEFILE "C:\SomeNewFile.txt",#221 1.23 2.54 8.22 -1.22<EOI> creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string of 21 characters " 1.23 2.54 8.22 -1.22". This is an example of using the definite length arbitrary block.

FILESYSTEM:WRITEFILE "C:\SomeNewFile.txt",#0 This is a test<NL+EOI> creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string "This is a test." This is an example of using the indefinite length arbitrary block. The newline character (0x0a) is required.

FILESYSTEM:WRITEFILE "C:\SomeNewFile.txt",#221 1.23 2.54 8.22 -1.22<NL+EOI> creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string of 21 characters " 1.23 2.54 8.22 -1.22". Because the data argument is indefinite format arbitrary block program data, it may contain any 8-bit values, including newline characters (hexadecimal 0a). The fact that newline characters are allowed makes this command deviate from the strictest interpretation of the IEEE488.2 specification.

## HARDCopy

This command sends a copy of the screen display to the port specified by HARDCopy:PORT. This command is equivalent to pressing the PRINT button on the front panel. When printing to a file, the file format is Windows Bitmap Image (BMP).

The HARDCopy query returns the port and file path.

**Group** Hardcopy

**Syntax** HARDCopy START  
HARDCopy?

**Related Commands** \*WAI, \*CLS

**Arguments** START initiates a screen copy to a file or the default system printer, as specified by the :HARDCopy:PORT selection. The default system printer is set within the Windows operating system. If you need information about how to set the default system printer, refer to Microsoft Windows online help.

---

**NOTE.** Use the \*WAI command between HARDCopy START commands to ensure that the first hard copy is complete before starting another.

---

**Examples** HARDCOPYSTART initiates a screen copy to a file or the default system printer, as specified by the :HARDCopy:PORT selection.

HARDCOPY? returns the hard copy file path and the selected port. The query might return :HARDCOPY:FILENAME &#x201c;C:\TekScope\Screen Captures\hcFILE.BMP&#x201d;;PORT FILE.

## HARDCopy:FILENAME

This command sets or queries the file that will be sent hardcopy data on the next HARDCopy command (if the HARDCopy:PORT is set to FILE).

**Group** Hardcopy

**Syntax** HARDCopy:FILENAME <file path>  
HARDCopy:FILENAME?

**Related Commands** [HARDCopy](#)

**Arguments** <file path> specifies that the hard copy is sent to the named file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. The file path cannot exceed 128 characters. If you don't specify the path with <drive>: and one or more <dir>s, the default location is "C:\TekScope\Screen Captures". While filename extensions are not required, they are highly recommended.

**Examples** HARDCOPY:FILENAME"TEK.BMP" sets TEK.BMP as the selected file name. The next hard copy will be sent to this file in the default hard copy directory, if the :HARDCopy:PORT selection is FILE.

HARDCOPY:FILENAME? might return :HARDCOPY:FILENAME "C:\TekScope\Screen Captures\TEK.BMP" as the selected file.

## HARDCopy:LAYout

This command sets or queries the page orientation for hard copy.

**Group** Hardcopy

**Syntax**    `HARDCopy:LAYout {PORTRait|LANDscape}`  
`HARDCopy:LAYout?`

**Arguments**    `PORTRait` orients the screen image vertically on the printed page.  
`LANDscape` orients the screen image horizontally on the printed page.

**Examples**    `HARDCOPY:LAYOUTLANDSCAPE` sets the hard copy page orientation to Landscape.  
`HARDCOPY:LAYOUT?` might return `:HARDCOPY:LAYOUT PORTRAIT`, indicating that the hard copy page orientation is set to portrait.

## HARDCopy:PALETTE

This command sets or queries the hard copy color palette.

**Group**    Hardcopy

**Syntax**    `HARDCopy:PALETTE {COLOR|INKSaver|BLACKANDwhite}`  
`HARDCopy:PALETTE?`

**Arguments**    `COLOR` argument sets the hard copy output to color.  
`INKSaver` argument sets the hard copy output to save ink.  
`BLACKANDwhite` argument sets the hard copy output to black and white.

**Examples**    `HARDCOPY:PALETTECOLOR` sets the hard copy output to Color.  
`HARDCOPY:PALETTE?` might return `:HARDCOPY:PALETTE GRAYSCALE`, indicating that the output for hard copy will be in gray scale.

## HARDCopy:PORT

This command selects or returns whether the hard copy data will be sent to a file or printed on the next hard copy command (for example, the `HARDCopy START` command). This is equivalent to selecting Print in the File menu and then either choosing Print to file or specifying the default printer. If FILE is selected then the EXPORT setup commands are used to create the file. If PRINTER is selected then the HARDCOPY commands are used to send the data to the printer.

**Group**    Hardcopy

**Syntax**     `HARDCopy:PORT {FILE|PRINTER}`  
`HARDCopy:PORT?`

**Related Commands**     [HARDCopy](#), [HARDCopy:FILENAME](#)

**Arguments**     `FILE` argument specifies that the hard copy is stored in the file specified in the [HARDCopy:FILENAME](#) command.

`PRINTER` argument specifies that the hard copy is sent to the printer specified in the Print dialog box.

**Examples**     `HARDCOPY:PORTFILE` specifies that the hard copy is stored in the file specified in the [HARDCopy:FILENAME](#) command.

`HARDCOPY:PORT?` might return `:HARDCOPY:PORT FILE` indicating that hard copies are sent to the specified file.

The following commands sequence lets you to specify an inksaver data hard copy is stored in a file:

```
*RST
HARDCOPY:PORT FILE
HARDCOPY:PALETTE INKSAVER
HARDCOPY:FILENAME "FILE_INKSAVER"
HARDCOPY START
```

The following commands sequence lets you to specify a color data hard copy is stored in a file:

```
HARDCOPY:PALETTE COLOR
HARDCOPY:FILENAME :FILE COLOR"
HARDCOPY START
```

Both files will be the same since they are directed to a file. If they were directed to a printer then the printouts would be different.

## HARDCopy:READOuts

This command sets or queries the area on a hardcopy where the readout appear. This command is equivalent to selecting Page Setup from the File menu and setting the Readouts Below Graticule in the control window.

**Group**     Hardcopy

**Syntax**     `HARDCopy:READOuts {OFFGRaticule|ONGRaticule}`  
`HARDCopy:READOuts?`



- Arguments** OFFGRATICULE places the readouts off the graticule area.  
 ONGRATICULE places the readouts on the graticule area on hardcopies.

## HARDCopy:VIEW

This command sets or queries the area of the screen to be hard copied.

**Group** Hardcopy

**Syntax** HARDCopy:VIEW {FULLSCREEN|GRATICULE|FULLNOMENU}  
 HARDCopy:VIEW?

- Arguments** FULLSCREEN sets the area to be hard copied to both the graticule and menu areas of the screen.  
 GRATICULE sets the area to be hard copied to only the graticule area of the screen.  
 FULLNOMENU sets the area to be hard copied to full screen but hides any menus or toolbars.

**Examples** HARDCOPY:VIEWFULLSCREEN sets the hard copy area to Full Screen.  
 HARDCOPY:VIEW? might return :HARDCOPY:VIEW GRATICULE, indicating that only the graticule area of the screen will be hard copied.

## HDR

This command is identical to the HEADer query and is included for backward compatibility purposes.

**Group** Miscellaneous

**Syntax** HDR {OFF|ON|<NR1>}  
 HDR?

**Related Commands** [HEADer](#), [VERBose](#)

- Arguments** 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.

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the instrument to send headers on query responses.

**Examples** HDROFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HDR? might return :HEADER 1, indicating that the instrument is including headers on applicable query responses.

## 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. This command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.*

---

**Group** Miscellaneous

**Syntax** HEADer {OFF|ON|<NR1>}  
HEADer?

**Related Commands** [HDR](#), [VERBose](#)

**Arguments** 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.

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true.

**Examples** HEADEROFF 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.

## HIStogram? (Query Only)

This query-only command returns all histogram parameters; it queries the state of all histogram parameters that the user can set. This command is equivalent to selecting Waveform Histograms from the Measure menu.

**Group** Histogram

**Syntax** HIStogram?

**Related Commands** [MEASUrement:METhod](#)

**Examples** HISTOGRAM? might return the histogram parameters as :HISTOGRAM:BOXPCNT 30.0000,25.1000,70.0000,75.2000;DISPLAY LINEAR;STATE 1;FUNCTION VERTICAL;SIZE 2.0000;SOURCE CH1.

## HIStogram:BOX

This command defines or returns the left, top, right, and bottom boundaries of the histogram box, in source waveform coordinates. This command is equivalent to selecting Waveform Histograms from the Measure menu and then setting Limits for Left, Right, Top, and Bottom in the appropriate boxes. The command is similar to the [HIStogram:BOXPent](#) command except that command uses percentage coordinates to define the boundaries of the histogram box.

---

**NOTE.** This command implicitly clears the histogram count and statistics for the histogram and histogram source data when HIStogram:STATE is turned on.

---

**Group** Histogram

**Syntax** HIStogram:BOX <NR3>,<NR3>,<NR3>,<NR3>  
HIStogram:BOX?

**Related Commands** [HIStogram:BOXPent](#)

**Arguments** <NR3> specifies the left position of the histogram box in source waveform coordinates.

<NR3> specifies the top position of the histogram box in source waveform coordinates.

<NR3> specifies the right position of the histogram box in source waveform coordinates.

<NR3> specifies the bottom position of the histogram box in source waveform coordinates.

**Examples** HISTOGRAM:BOX1E-9, 0.250, 2E-9, 0.500 defines the coordinates of the histogram box in source waveform coordinates.

HISTOGRAM:BOX? might return :HISTOGRAM:BOX 1.0000E-09, 0.2500, 2.0000E-09, 0.5000, which identifies the coordinates of the histogram box in source waveform coordinates.

## HIStogram:BOXPcnt

This command defines or queries the left, top, right, and bottom positions of the histogram box, in percentage coordinates. The upper left has the value 0,0 and the lower right has the value 100, 100. Any value outside the range of 0 to 100 is clipped. This command is similar to the [HIStogram:BOX](#) command except that this command uses percentage coordinates to define the boundaries of the histogram box.

---

**NOTE.** This command implicitly clears the histogram count and statistics for the histogram and histogram source data when *HIStogram:STATE* is turned on.

---

**Group** Histogram

**Syntax** HIStogram:BOXPcnt <NR2>, <NR2>, <NR2>, <NR2>  
 HIStogram:BOXPcnt?

**Related Commands** [HIStogram:BOX](#)

**Arguments** <NR2> specifies the left position of the histogram box in percentage coordinates. The default value is 30.0.

<NR2> specifies the top position of the histogram box in percentage coordinates. The default value is 25.1.

<NR2> specifies the right position of the histogram box in percentage coordinates. The default value is 70.0.

<NR2> specifies the bottom position of the histogram box in percentage coordinates. The default value is 75.2.

**Examples** HISTOGRAM:BOXPCNT30.0,25.0,70.0,75.0 sets the left boundary of the histogram box to 30% of the graticule (3 divisions from the left edge); the top boundary to 25% of the graticule ( $0.25 \times 8 = 2.0$  divisions from the top edge); the right boundary to 70% of the graticule (7 divisions from the left edge); and the bottom boundary to 75% of the graticule ( $0.75 \times 8 = 6.0$  divisions from the top edge).

HISTOGRAM:BOXPCNT? might return :HISTOGRAM:BOXPCNT  
30.0000,25.1000,70.0000,75.2000.

## HISTogram:COUNT (No Query Form)

This command (no query form) clears the count and statistics for the histogram and the histogram source data. If the histogram is on, then counting restarts. This command is equivalent selecting Waveform Histograms from the Measure menu and then clicking the Reset button.

**Group** Histogram

**Syntax** HISTogram:COUNT RESET

**Related Commands** [HISTogram:BOX](#), [HISTogram:BOXPent](#)

**Examples** HISTOGRAM:COUNTRRESET clears the count and statistics for the histogram and the histogram source data.

## HISTogram:DATA? (Query Only)

This query-only command returns an ASCII comma-separated list of histogram data numbers. This command returns 252 values for vertical histograms and 1000 values for horizontal histograms. The first and last value are not displayed, and are used for over range and under range tracking.

---

**NOTE.** *You must set the histogram state to ON before sending this query or the system will generate an error indicating: "Source waveform is not active".*

---



<b>Group</b>	Histogram
<b>Syntax</b>	HISTogram:DISPlay {OFF LOG LINEAr} HISTogram:DISPlay?
<b>Related Commands</b>	<a href="#">HISTogram:MODE</a> , <a href="#">HISTogram:STATE</a>
<b>Arguments</b>	<p>&lt;OFF&gt; disables the histogram display but allows histogram data to accumulate. This argument provides for backward compatibility with older DPO Series instruments; it does not affect a change in the display scaling of the histogram. When queried, HISTogram:DISPlay? will return OFF if the histogram state or mode has been disabled using the HISTogram:MODE or HISTogram:STATE commands.</p> <p>&lt;LOG&gt; sets the histogram display to logarithmic scaling.</p> <p>&lt;LINEAR&gt; sets the histogram display to linear scaling. This is the default setting.</p>
<b>Examples</b>	<p>HISTOGRAM:DISPLAYLOG sets the histogram scaling to be the log count of each histogram bin.</p> <p>HISTOGRAM:DISPLAY? might return :HISTOGRAM:DISPLAY LOG indicating that the current histogram display scaling is the log count of each histogram bin.</p>

## HISTogram:FUNCTION

This command either selects the type of histogram to create or returns the current histogram type. This command is equivalent to selecting Waveform Histogram from the Measure menu and then choosing either Horizontal or Vertical from the Histogram Mode group box.

<b>Group</b>	Histogram
<b>Syntax</b>	HISTogram:FUNCTION {HORIZONTAL VERTICAL} HISTogram:FUNCTION?
<b>Related Commands</b>	<a href="#">HISTogram:MODE</a> , <a href="#">HISTogram:STATE</a>
<b>Arguments</b>	<p>HORIZONTAL displays a horizontally positioned histogram that shows time distribution.</p> <p>VERTICAL displays a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).</p>

**Examples** HISTOGRAM:FUNCTIONHORIZONTAL sets the histogram mode to the horizontal display mode.

HISTOGRAM:FUNCTION? might return :HISTOGRAM:FUNCTION VERTICAL, indicating that the vertical display mode is selected for the histogram.

## HIStogram:MODe

This command selects the type of histogram to create or disables the histogram display. The query form either returns the current histogram type or that histogram display is disabled. This command is equivalent to selecting Waveform Histogram from the Measure menu and then choosing from the Histogram Mode group box.

HIStogram:MODe is functionally equivalent to the [HIStogram:FUNcTion](#) and [HIStogram:STATe](#) commands and is included for backward compatibility purposes.

**Group** Histogram

**Syntax** HIStogram:MODe {HORizontal|VERTical|OFF}  
HIStogram:MODe?

**Related Commands** [HIStogram:FUNcTion](#), [HIStogram:STATe](#), [HIStogram:DISplay](#)

**Arguments** HORizontal enables a horizontally positioned histogram that shows time distribution.

VERTical enables a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).

OFF disables collection of the histogram measurement.

**Examples** HISTOGRAM:MODEHORIZONTAL sets the type of histogram created to horizontal, which displays a horizontally positioned histogram that shows time distribution.

HISTOGRAM:MODE? might return :HISTOGRAM:MODE OFF, indicating that histogram display is disabled.

## HIStogram:SIZe

This command sets or queries the width or height of the histogram on the screen. This is equivalent to selecting Waveform Histograms from the Measure menu and then entering a value in the Histogram Size box.



<b>Group</b>	Histogram
<b>Syntax</b>	HISTogram:SIZE <NR3> HISTogram:SIZE?
<b>Related Commands</b>	<a href="#">HISTogram:DISplay</a>
<b>Arguments</b>	<NR3> specifies the histogram size. The value can vary from 0.1 to 8.0 divisions in HORizontal mode and from 0.1 to 10.0 divisions in VERTical mode. Resolution is to the nearest pixel.
<b>Examples</b>	HISTOGRAM:SIZE2.0 sets the size of the histogram to 2.0 divisions.  HISTOGRAM:SIZE? might return :HISTOGRAM:SIZE 2.0000E+01, indicates the histogram size is set to 2.0 divisions.

## HISTogram:SOURce

This command sets or queries which source will be compared against the histogram box when the histogram testing is enabled. This is equivalent to selecting Waveform Histograms from the Measure menu and then choosing the desired waveform source. The waveform need not be displayed for histograms to run. You might want the channel displays disabled so you can see a full-screen histogram and not have waveform data confuse the display.

<b>Group</b>	Histogram
<b>Syntax</b>	HISTogram:SOURce {CH<x> MATH<x> REF<x>} HISTogram:SOURce?
<b>Related Commands</b>	<a href="#">HISTogram:DISplay</a>
<b>Arguments</b>	CH<x> selects a channel waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4.  MATH<x> selects a math waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4.  REF<x> selects a reference waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4.

**Examples** HISTOGRAM:SOURCECH1 enables the Channel 1 waveform to be compared against the histogram box. The default time base, Main, is used for the histogram display.

HISTOGRAM:SOURCE? might return :HISTOGRAM:SOURCE CH1, indicating that the waveform for Channel 1 is the source for the histogram.

## HISTogram:STATE

This command sets or queries whether the histogram calculations are enabled. This is equivalent to selecting Waveform Histograms from the Measure menu.

**Group** Histogram

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

**Related Commands** [HISTogram:MODE](#), [MEASUrement:METHOD](#)

**Arguments** <ON> enables histogram calculations.  
<OFF> disables the histogram calculations.  
<NR1> = 0 disables histogram calculations; any other value enables histogram calculations.

**Examples** HISTOGRAM:STATEON enables histogram calculations.

HISTOGRAM:STATE? might return :HISTOGRAM:STATE 0, indicating that histogram calculations are currently disabled.

## HORizontal? (Query Only)

This query-only command returns all settings for the horizontal commands.

**Group** Horizontal

**Syntax** HORizontal?

**Examples** HORIZONTAL? might return the following horizontal settings  
:HORIZONTAL:DELAY:MODE 0;POSITION 50.0000;TIME  
0.000000000000;:HORIZONTAL:MAIN:INTERPRATIO

```
1.0000;POSITION 50.0000;UNITS:STRING "s";:HORIZONTAL:ROLL
AUTO;MODE AUTO;MODE:AUTO:LIMITRECORDLEN
200000;:HORIZONTAL:MODE:RECORDLENGTH 1000;SAMPLERATE
40.0000E+9;SCAL.
```

## 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:ACQLENGTH? (Query Only)

This query returns the record length.

**Group** Horizontal

**Syntax** HORizontal:ACQLENGTH?

**Returns** <NR1> returns the record length.

**Examples** HORIZONTAL:ACQLENGTH? might return :HORIZONTAL:ACQLENGTH 50000, indicating the acquisition record length is 50000 points.

## HORizontal:DIGital:RECOrdlength? (Query Only)

This command queries the record length of the digital acquisition.

**Group** Horizontal

**Syntax**    `HORizontal:DIGital:RECOrdlength?`

**Related Commands**    [HORizontal:DIGital:SAMPLERate?](#)

**Examples**    `HORIZONTAL:DIGITAL:RECORDLENGTH?` might return  
`:HORIZONTAL:DIGITAL:RECORDLENGTH 125000`, indicating that  
the record length of the horizontal digital acquisition is set to 125000.

## **HORizontal:DIGital:SAMPLERate? (Query Only)**

This command queries the sample rate of the digital acquisition.

**Group**    Horizontal

**Syntax**    `HORizontal:DIGital:SAMPLERate?`

**Related Commands**    [HORizontal:DIGital:RECOrdlength?](#)

**Examples**    `HORIZONTAL:DIGITAL:SAMPLERATE?` might return  
`:HORIZONTAL:DIGITAL:SAMPLERATE 62.5000E+6`, indicating that the  
sample rate of the horizontal digital acquisition is set to 62.5000E+6.

## **HORizontal:DIVisions? (Query Only)**

This query-only command returns the number of graticule divisions over which  
the waveform is displayed.

**Group**    Horizontal

**Syntax**    `HORizontal:DIVisions?`

**Examples**    `HORIZONTAL:DIVISIONS?` might return `:HORIZONTAL:DIVISIONS 10.0000`,  
indicating that the waveform is displayed across ten division.

## HORizontal:FASTframe? (Query Only)

This query-only command returns all of the settings for the horizontal FastFrame commands.

**Group** Horizontal

**Syntax** HORizontal:FASTframe?

**Examples** HORIZONTAL:FASTFRAME? might return :HORIZONTAL:FASTFRAME:STATE 1;READOUTS0;SUMFRAME NONE;MULTIPLEFRAMES:MODE OVERLAY;FRAMESTART:CH1 1;CH2 1;CH3 1;CH4 1;MATH1 1;MATH2 1;MATH3 1;MATH4 1;REF1 1;REF2 1;REF3 1;REF4 1;:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUMFRAMES:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME:LENGTH 500;COUNT 2;SELECTED:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME:REF:SOURCE CH4;FRAME 1;:HORIZONTAL:FASTFRAME:TRACK ALL.

## HORizontal:FASTframe:COUNT

This command sets or queries FastFrame frame count. This command is equivalent to selecting Fast Frame Setup from the Horiz/Acq menu and entering a value in the Frame Count box. FastFrame captures a series of triggered acquisitions with minimal intervening time.

**Group** Horizontal

**Syntax** HORizontal:FASTframe:COUNT <NR1>  
HORizontal:FASTframe:COUNT?

**Related Commands** [HORizontal:MODE:RECOrdlength](#)

**Arguments** <NR1> indicates the number of frames to acquire.

**Examples** HORIZONTAL:FASTFRAME:COUNT2 sets up FastFrame mode to acquire two frames (segments) of data.

HORIZONTAL:FASTFRAME:COUNT? might return  
:HORIZONTAL:FASTFRAME:COUNT 4, indicating that FastFrame mode is  
currently set to acquire 4 frames of data.

## HORizontal:FASTframe:MAXFRames? (Query Only)

This query only command returns the maximum number of FastFrame frames  
which can be acquired at the current frame length.

<b>Group</b>	Horizontal
<b>Syntax</b>	HORizontal:FASTframe:MAXFRames?
<b>Examples</b>	HORIZONTAL:FASTFRAME:MAXFRAMES? might return HORIZONTAL: FASTFRAME:MAXFRAMES 3, indicating that the maximum number of FastFrame frames which can be acquired at the current frame length is 3.

## HORizontal:FASTframe:MULTipleframes:FRAMEStart:<wfm>

This command sets or queries the start frame number on the specified waveform  
for the FastFrame multiple frames feature. Valid waveforms include CH<x>,  
MATH<x>, and REF<x>. The value of x can be an integer ranging from 1  
through 4. The multiple frames feature supports displaying multiple frames in an  
overlaid manner.

<b>Group</b>	Horizontal
<b>Syntax</b>	HORizontal:FASTframe:MULTipleframes:FRAMEStart:<wfm> <NR1> HORizontal:FASTframe:MULTipleframes:FRAMEStart:<wfm>?
<b>Related Commands</b>	<a href="#">HORizontal:FASTframe:MULTipleframes:NUMFRames:&lt;wfm&gt;</a> , <a href="#">HORizontal: FASTframe:MULTipleframes:MODE</a> , <a href="#">HORizontal:FASTframe:COUNT</a>
<b>Arguments</b>	<NR1> specifies the start frame number on the specified waveform.
<b>Examples</b>	HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:FRAMESTART:CH1 3 sets the start frame number to frame 3 on Channel 1.

HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:FRAMESTART:CH1? might return :HORIZONTAL:FASTFRAME:MULTIFRAMES:FRAMESTART:CH1 3, indicating that the start frame number is frame 3 for Channel 1.

## HORizontal:FASTframe:MULTipleframes:MODE

This command sets or queries the mode for the FastFrame multiple frames feature. This feature displays multiple frames in an overlaid manner.

**Group** Horizontal

**Syntax** HORizontal:FASTframe:MULTipleframes:MODE  
{OFF|OVERlay|ONLYOVERlay}  
HORizontal:FASTframe:MULTipleframes:MODE?

**Related Commands** [HORizontal:FASTframe:MULTipleframes:FRAMEStart:<wfm>](#),  
[HORizontal:FASTframe:MULTipleframes:NUMFRames:<wfm>](#),  
[HORizontal:FASTframe:COUNt](#)

**Arguments** OFF turns off the multiple frames mode.  
OVERlay sets the multiple frames mode to overlay.

**Examples** HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:MODEOFF turns off the multiple frames mode.  
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:MODE? might return :HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:MODE OFF, indicating that the multiple frames mode is off.

## HORizontal:FASTframe:MULTipleframes:NUMFRames:<wfm>

This command sets or queries the number of frames on the specified waveform for the FastFrame multiple frames feature. The multiple frames feature supports displaying multiple frames in an overlaid manner.

Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

**Group** Horizontal

<b>Syntax</b>	<code>HORizontal:FASTframe:MULTipleframes:NUMFRames:&lt;wfm&gt; &lt;NR1&gt;</code> <code>HORizontal:FASTframe:MULTipleframes:NUMFRames:&lt;wfm&gt;?</code>
<b>Related Commands</b>	<a href="#">HORizontal:FASTframe:MULTipleframes:FRAMEStart:&lt;wfm&gt;</a> , <a href="#">HORizontal:FASTframe:MULTipleframes:MODE</a> , <a href="#">HORizontal:FASTframe:COUNt</a>
<b>Arguments</b>	<NR1> represents the number of frames on the specified waveform.
<b>Examples</b>	<code>HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUMFRAMES:CH1 3</code> sets the number of frames to 3 on Channel 1.  <code>HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUMFRAMES:CH1?</code> might return <code>:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUMFRAMES:CH1 1</code> , indicating that the number of frames on Channel 1 is 3.

## HORizontal:FASTframe:REF:FRame

This command sets or queries the FastFrame reference frame number. The Reference Frame number is used to calculate time differences for each frame and displays those differences in the graticule. This command is equivalent to selecting FastFrame Setup in the Horiz/Acq menu and entering a value in the Time Stamps/Frame box.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal:FASTframe:REF:FRame &lt;NR1&gt;</code> <code>HORizontal:FASTframe:REF:FRame?</code>
<b>Related Commands</b>	<a href="#">HORizontal:FASTframe:REF:SOURce</a> , <a href="#">HORizontal:FASTframe:TIMEStamp:DELTA:&lt;wfm&gt;?</a> , <a href="#">HORizontal:FASTframe:TIMEStamp:REF?</a> , <a href="#">HORizontal:FASTframe:TRACk</a>
<b>Arguments</b>	<NR1> specifies the FastFrame reference frame.
<b>Examples</b>	<code>HORIZONTAL:FASTFRAME:REF:FRAME2</code> sets the FastFrame reference frame to 2.  <code>HORIZONTAL:FASTFRAME:REF:FRAME?</code> might return <code>:HORIZONTAL:FASTFRAME:REF:FRAME 3</code> , indicating that the FastFrame reference frame is currently set to 3.



## HORizontal:FASTframe:REF:SOURce

This command sets or queries FastFrame Reference waveform source. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and choosing the reference source.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:REF:SOURce <wfm>`  
`HORizontal:FASTframe:REF:SOURce?`

**Related Commands** [HORizontal:FASTframe:REF:FRAMe](#), [HORizontal:FASTframe:TIMEStaMp:DELTA:<wfm>?](#), [HORizontal:FASTframe:TIMEStaMp:REF?](#), [HORizontal:FASTframe:TRACk](#)

**Arguments** <wfm> specifies the FastFrame Reference waveform source. Valid waveforms include CH<x> and MATH<x> The x variable can be expressed as an integer ranging from 1 through 4.

**Examples** `HORIZONTAL:FASTFRAME:REF:SOURCECH2` sets the horizontal FastFrame waveform reference source to Channel 2.

`HORIZONTAL:FASTFRAME:REF:SOURCE?` might return `:HORIZONTAL:FASTFRAME:REF:SOURCE CH2`, indicating that the FastFrame waveform source is currently set to Channel 2.

## HORizontal:FASTframe:SELECTED:CH<x>

This command sets or queries the FastFrame selected frame number on the specified waveform.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:SELECTED:CH<x> <NRF>`  
`HORizontal:FASTframe:SELECTED:CH<x>?`

**Arguments** <NRF> is the selected frame number.

**Examples**     `HORIZONTAL:FASTFRAME:SELECTED:CH1 5` selects frame 5 on Channel 1.  
`HORIZONTAL:FASTFRAME:SELECTED:CH1?` might return  
`:HORIZONTAL:FASTFRAME:SELECTED:CH1 5`, indicating frame 5  
on Channel 1 is selected.

## **HORizontal:FASTframe:SELECTED:MATH<x>**

This command sets or queries the FastFrame selected frame number on the specified waveform.

**Group**     Horizontal

**Syntax**     `HORizontal:FASTframe:SELECTED:MATH<x> <NRF>`  
`HORizontal:FASTframe:SELECTED:MATH<x>?`

**Arguments**     `<NRF>` is the selected frame number.

**Examples**     `HORIZONTAL:FASTFRAME:SELECTED:MATH2 4` selects frame 4 on Math 2  
waveform.  
`HORIZONTAL:FASTFRAME:SELECTED:MATH1?` might return  
`:HORIZONTAL:FASTFRAME:SELECTED:MATH1 5`, indicating that frame 5 on  
Math 1 waveform is selected.

## **HORizontal:FASTframe:SELECTED:REF<x>**

This command sets or queries the FastFrame selected frame number on the specified waveform.

**Group**     Horizontal

**Syntax**     `HORizontal:FASTframe:SELECTED:REF<x> <NRF>`  
`HORizontal:FASTframe:SELECTED:REF<x>?`

**Arguments**     `<NRF>` is the selected frame number.

**Examples**     `HORIZONTAL:FASTFRAME:SELECTED:REF1 1` selects frame 1 on Reference  
1 waveform.

HORIZONTAL:FASTFRAME:SELECTED:REF1? might return  
:HORIZONTAL:FASTFRAME:SELECTED:REF1 25, indicating frame 25 on  
Reference 1 waveform is the selected frame.

## HORizontal:FASTframe:SELECTED:SOURce

This command sets or queries the FastFrame source waveform. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu, and then choosing the waveform source. Valid waveforms include CH<x> and MATH<x>. The x variable can be expressed as an integer ranging from 1 through 4.

**Group** Horizontal

**Syntax** HORIZONTAL:FASTframe:SELECTED:SOURce <NR1>  
HORIZONTAL:FASTframe:SELECTED:SOURce?

**Related Commands** HORizontal:FASTframe:TRACk, HORizontal:FASTframe:TIMEStamp:DELTA:  
<wfm>?, HORizontal:FASTframe:TIMEStamp:REF?

**Arguments** <NR1> specifies the selected frame number on the specified waveform.

**Examples** HORIZONTAL:FASTFRAME:SELECTED:SOURCE CH2 sets the FastFrame source  
to Channel 2.

HORIZONTAL:FASTFRAME:SELECTED:SOURCE? might return  
:HORIZONTAL:FASTFRAME:SELECTED:SOURCE CH1, indicating that the  
selected source is Channel 1.

## HORizontal:FASTframe:SEQuence

This command sets or queries the FastFrame single-sequence mode stop condition.

**Group** Horizontal

**Syntax** HORIZONTAL:FASTframe:SEQuence {FIRST|LAST}  
HORIZONTAL:FASTframe:SEQuence?

**Arguments** FIRST sets single sequence to stop after n frames.

LAST sets single sequence to stop manually.

**Examples**     `HORIZONTAL:FASTFRAME:SEQUENCEFirst` sets single-sequence mode to stop after `n` frames.

`HORIZONTAL:FASTFRAME:SEQUENCE?` might return  
`:HORIZONTAL:FASTFRAME:SEQUENCE LAST`, indicating the FastFrame single-sequence mode is stop manually.

## HORizontal:FASTframe:STATE

This command sets or queries the state of FastFrame acquisition. This is equivalent to setting FastFrame to On in the FastFrame Setup menu. FastFrame lets users capture a series of triggered acquisitions with minimal time between them.

The digitizing instrument in FastFrame mode is ready to accept a continuous burst of triggers 400 ms after the controller sends the [ACQUIRE:STATE RUN](#) command.

**Group**     Horizontal

**Syntax**     `HORizontal:FASTframe:STATE {ON|OFF|<NR1>}`  
`HORizontal:FASTframe:STATE?`

**Related Commands**     [ACQUIRE:STATE](#)

**Arguments**     `<ON>` turns on horizontal FastFrame.  
`<OFF>` turns off horizontal FastFrame.  
`<NR1> = 0` turns off horizontal FastFrame; any other value turns on horizontal FastFrame.

**Examples**     `HORIZONTAL:FASTFRAME:STATEON` turns on horizontal FastFrame.

`HORIZONTAL:FASTFRAME:STATE?` might return  
`:HORIZONTAL:FASTFRAME:STATE 0`, indicating that the current state of FastFrame is off.

## HORizontal:FASTframe:SUMFrame

This command sets or queries the summary frame mode. When ENvelope is selected, the last frame in a FastFrame acquisition is an envelope of all the prior frames in the acquisition. When AVERAGE is selected, the last frame is replaced with a frame that is the computed average of all the prior frames in the acquisition.

For the summary frame control to be active, the number of frames must be two or greater.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:SUMFrame {NONE|ENVELOPE|AVERAGE}`  
`HORizontal:FASTframe:SUMFrame?`

**Arguments** NONE turns off the summary mode for FastFrame. This is the default setting.

ENVELOPE argument displays the last frame in a FastFrame acquisition as an envelope of all the prior frames in the acquisition.

AVERAGE argument displays the last frame in a FastFrame acquisition as a frame that is the computed average of all the prior frames in the acquisition.

**Examples** `HORIZONTAL:FASTFRAME:SUMFRAMENONE` turns off the summary frame feature.

`HORIZONTAL:FASTFRAME:SUMFRAME?` might return  
`:HORIZONTAL:FASTFRAME:SUMFRAME AVERAGE`, indicating that the summary mode for FastFrame is set to AVERAGE and will display the last frame in the acquisition as the average of all the prior frames in the acquisition.

## **HORizontal:FASTframe:TIMESTAMP:ALL:<wfm>? (Query Only)**

This query-only command returns the frame number and time stamp for each frame between requested frames, inclusive, within the specified waveform. Argument order is unimportant. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

The format of the output is as follows:

`DD MON YYYY HH:MM:SS.xxx xxx xxx xxx`

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:TIMESTAMP:ALL:<wfm>? <NR1>, <NR1>`

**Arguments** <NR1> specifies the selected “from” frame number within the specified waveform.  
 <NR1> specifies the selected “to” frame number within the specified waveform.

**Examples** HORIZONTAL:FASTFRAME:TIMESTAMP:ALL:CH2? 4,1 might return  
 :HORIZONTAL:FASTFRAME:TIMESTAMP:ALL:CH1 4,1,“02 Mar 2000  
 20:10:54.542 037 272 620” This is the list of time stamps from frame 1 through  
 frame 4 on Channel 1.

## HORizontal:FASTframe:TIMEStamp:BETween:<wfm>? (Query Only)

This query-only command returns the relative trigger for the delta time between the specified frames, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

The format of the delta time output is as follows:

HH:MM:SS.xxx xxx xxx xxx

- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** HORizontal:FASTframe:TIMEStamp:BETween:<wfm>? <NR1>, <NR1>

**Arguments** <NR1> specifies the selected “from” frame number within the specified waveform.  
 <NR1> specifies the selected “to” frame number within the specified waveform.

**Examples** HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN:CH1? 5,2 might  
 return :HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN:CH1  
 5,2,“00:00:00.010 000 540 624” This is the delta of Channel 1’s Time  
 Stamp frame 5 - Time Stamp frame 2 (TS[5]-TS[2]).

## HORizontal:FASTframe:TIMEStamp:DELTA:<wfm>? (Query Only)

This query-only command returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform. Valid waveforms include CH<x>, MATH<x>, and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

The format of the output is as follows:

HH:MM:SS.xxx xxx xxx xxx

- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** HOrizontal:FASTframe:TIMESTamp:DELTA:<wfm>?

**Examples** HORIZONTAL:FASTRAME:TIMESTAMP:DELTA:CH1? might return  
:HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA:CH1 "00:00:00.006 000  
306 556" This is the delta time stamp of FastFrame Selected Frame on Channel 1  
- FastFrame Reference Frame (TS[C1 Selected]-TS[Reference]).

## HORizontal:FASTframe:TIMESTamp:FRAME:<wfm>? (Query Only)

This query-only command returns the absolute trigger date and time for the specified frame and waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** HOrizontal:FASTframe:TIMESTamp:FRAME:<wfm>? <NR1>

**Arguments** <NR1> is the frame number for which the timestamp will be returned on the specified waveform.

**Examples** HORIZONTAL:FASTFRAME:TIMESTAMP:FRAME:CH1? 10 might return  
:HORIZONTAL:FASTFRAME:TIMESTAMP:FRAME:CH1 10,"02 Mar 2000  
20:10:54.536 036 928 432" which is the time stamp of Channel 1, frame 10.

## HORizontal:FASTframe:TIMEStamp:REF? (Query Only)

This query-only command returns the absolute trigger date and time for FastFrame reference.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** HORizontal:FASTframe:TIMEStamp:REF?

**Related Commands** [HORizontal:FASTframe:REF:FRAME](#), [HORizontal:FASTframe:REF:SOURce](#)

**Examples** HORIZONTAL:FASTFRAME:TIMESTAMP:REF? might return  
:HORIZONTAL:FASTFRAME:TIMESTAMP:REF "02 Mar 2000  
20:10:54.533 036 838 784" This is the time stamp of FastFrame Reference.

## HORizontal:FASTframe:TIMEStamp:SELECTED:<wfm>? (Query Only)

This query-only command returns the absolute trigger date and time for the FastFrame Selected, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

**Group** Horizontal

**Syntax** HORizontal:FASTframe:TIMEStamp:SELECTED:<wfm>?



**Related Commands** [HORizontal:FASTframe:SELECTED:SOUrce](#)

**Examples** `HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED:REF1?` might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED:REF1 "02 MAR 2000 20:10:54:539 037 145 340"` This is the time stamp of FastFrame Selected on Reference 1 waveform.

## HORizontal:FASTframe:TRACK

This command sets up or queries the state of FastFrame tracking feature. This command is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and then clicking the desired Frame Tracking state.

When FastFrame Track is set to “live”, the channel and math waveforms are locked together. Adjusting a channel waveform also adjusts a related math waveform. All reference waveforms are also locked together but they are separate from channel and math waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3 and Math4 are also set to Frame 3.

When you set the Selected Frame Source Ref1 to Frame 2, then Selected Frame Ref2, Ref3 and Ref4 are also set to Frame 2. If the Frame Tracking is set to Live, changing Selected Frame Ch1 will not affect the Selected Frame Ref1 frame of the Reference Frame setting.

When FastFrame Track is set to “all”, the channel, math and reference waveforms are locked together. Adjusting a channel waveform also adjusts the related math and reference waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3 and Ref4 are also set to Frame 3.

**Group** Horizontal

**Syntax** `HORIZONTAL:FASTframe:TRACK {LIVE|ALL}`  
`HORIZONTAL:FASTframe:TRACK?`

**Related Commands** [HORizontal:FASTframe:REF:FRaMe](#), [HORizontal:FASTframe:REF:SOUrce](#), [HORizontal:FASTframe:SELECTED:SOUrce](#)

**Arguments** LIVE sets FastFrame Track to Live.

ALL sets FastFrame Track to All.

**Examples** HORIZONTAL:FASTFRAME:TRACKLIVE sets FastFrame Track to live.

HORIZONTAL:FASTFRAME:TRACK? might return  
:HORIZONTAL:FASTFRAME:TRACK ALL, indicating that all FastFrame Selected  
Frames and the FastFrame Reference will be locked together.

## HORizontal:FASTframe:XZEro:ALL:CH<x>? (Query Only)

This command queries the time from the trigger to the trigger sample of the specified frames on the specified channel.

**Group** Horizontal

**Syntax** HORIZONTAL:FASTframe:XZEro:ALL:CH<x>? <NR1>, <NR1>

**Arguments** <NR1> specifies the first and last frame of a range of frames.

**Examples** HORIZONTAL:FASTFRAME:XZERO:ALL:CH1? 1,4 might  
return :HORIZONTAL:FASTFRAME:XZERO:ALL:CH1 1,4,"1:  
2.5265E-009","2: 3.9010E-010","3: 3.8285E-009","4:  
1.3237E-009", giving the first 4 XZERO times for Channel 1.

## HORizontal:FASTframe:XZEro:ALL:REF<x>? (Query Only)

This command queries the time from the trigger to the trigger sample of the specified frames on the specified reference.

**Group** Horizontal

**Syntax** HORIZONTAL:FASTframe:XZEro:ALL:REF<x>? <NR1>, <NR1>

**Arguments** <NR1> specifies the first and last frame of a range of frames.

**Examples** HORIZONTAL:FASTFRAME:XZERO:ALL:REF1? 1,3 might  
return :HORIZONTAL:FASTFRAME:XZERO:ALL:REF1 1,3,"1:

2.5265E-009", "2: 3.9010E-010", "3: 3.8285E-009", giving the first 3XZERO times for Reference 1.

## HORizontal:FASTframe:XZero:FRame:CH<x>? (Query Only)

This command queries the time from the trigger to the trigger sample of the specified frame on the specified channel.

**Group** Horizontal

**Syntax** HORizontal:FASTframe:XZero:FRame:CH<x>? <NR1>

**Arguments** <NR1> specifies a frame on the specified channel.

**Examples** HORIZONTAL:FASTFRAME:XZERO:FRAME:CH1? 3 might return  
:HORIZONTAL:FASTFRAME:XZERO:FRAME:CH1 3,3.8285E-9, indicating the time from the trigger to the start of frame 3 on Channel 1 is 3.83 ns.

## HORizontal:FASTframe:XZero:FRame:REF<x>? (Query Only)

This command queries the time from the trigger to the trigger sample of the specified frame on the specified reference.

**Group** Horizontal

**Syntax** HORizontal:FASTframe:XZero:FRame:REF<x>? <NR1>

**Arguments** <NR1> specifies a frame on the specified reference.

**Examples** HORIZONTAL:FASTFRAME:XZERO:FRAME:REF1? 3 might return  
:HORIZONTAL:FASTFRAME:XZERO:FRAME:REF1 3,3.8285E-9, indicating the time from the trigger to the start of frame 3 on Reference 1 is 3.83 ns.

## HORizontal:FASTframe:XZero:REF? (Query Only)

This command sets or queries the time from the trigger to the trigger sample on the reference waveform.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:XZero:REF?`

**Examples** `HORIZONTAL:FASTFRAME:XZERO:REF1?` might return  
`:HORIZONTAL:FASTFRAME:XZERO:REF 35.1000E-12`, indicating  
 that the Reference 1 waveform trigger sample occurred 35.1 ps after the trigger.

## **HORizontal:FASTframe:XZero:SELECTED:CH<x>? (Query Only)**

This command sets or queries the time from the trigger to the trigger sample on the selected channel. CH<x> can be CH1, CH2, CH3, or CH4.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:XZero:SELECTED:CH<x>?`

**Examples** `HORIZONTAL:FASTFRAME:XZERO:SELECTED:CH1?` might return  
`:HORIZONTAL:FASTFRAME:XZERO:SELECTED:CH1 35.1000E-12`, indicating  
 that the selected Channel 1 trigger sample occurred 35.1 ps after the trigger.

## **HORizontal:FASTframe:XZero:SELECTED:REF<x>? (Query Only)**

This command sets or queries the time from the trigger to the trigger sample on the selected reference waveform. REF<x> can be REF1, REF2, REF3, or REF4.

**Group** Horizontal

**Syntax** `HORizontal:FASTframe:XZero:SELECTED:REF<x>?`

**Examples** `HORizontal:FASTframe:XZero:SELECTED:REF1?` might return  
`:HORIZONTAL:FASTFRAME:XZERO:SELECTED:REF1 35.1000E-12`,  
 indicating that the selected Reference 1 waveform trigger sample occurred 35.1 ps  
 after the trigger.

## HORizontal:MAIn? (Query Only)

This query-only command returns the time per division of the time base. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu.

**Group** Horizontal

**Syntax** HORizontal:MAIn?

**Related Commands** [HORizontal:MODE:SCALE](#)

**Examples** HORIZONTAL:MAIN? might return :HORIZONTAL:MAIN:INTERPRATIO 1.0000;SCALE 40.0000E-9;POSITION 50.0000;SAMPLERATE 1.2500E+9;UNITS:STRING "s".

## HORizontal[:MAIn]:DELay:MODE

This command sets or queries the time base trigger delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/Acq menu.

**Group** Horizontal

**Syntax** HORizontal[:MAIn]:DELay:MODE {ON|OFF|<NR1>}  
HORizontal[:MAIn]:DELay:MODE?

**Related Commands** [HORizontal\[:MAIn\]:DELay:TIME](#)

**Arguments** ON enables the time base trigger delay mode.  
OFF disables the time base trigger delay mode.  
<NR1> = 0 disables the time base trigger delay mode, any other value enables the time base trigger delay mode.

**Examples** HORIZONTAL:DELAY:MODE ON enables the time base trigger delay mode.  
HORIZONTAL:DELAY:MODE? might return :HORIZONTAL:DELAY:MODE 1, indicating that the time base trigger delay mode is currently enabled.

## HORizontal[:MAIn]:DELay:POSition

This command sets or queries the time base position when Horizontal Delay Mode is turned on. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu and then entering a Ref Point value.

**Group** Horizontal

**Syntax** `HORizontal[:MAIn]:DELay:POSition <NR3>`  
`HORizontal[:MAIn]:DELay:POSition?`

**Related Commands** [HORizontal\[:MAIn\]:DELay:TIME](#)

**Arguments** <NR3> is from 0 to  $\approx 100$  and is the percentage of the waveform that is displayed left of the center graticule.

---

**NOTE.** *The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit =  $100 - 1/\text{record length}$ ).*

---

**Examples** `HORIZONTAL:DELAY:POSITION 5E+1` sets the time base trigger delay position to 50 percent.

`HORIZONTAL:DELAY:POSITION?` might return  
`:HORIZONTAL:MAIN:DELAY:POSITION 5.0000E+01`, indicating that the time base trigger delay position is currently set to 50 percent.

## HORizontal[:MAIn]:DELay:TIME

This command sets or queries the time base trigger delay time. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and choosing a value for Horiz Delay.

**Group** Horizontal

**Syntax** `HORizontal[:MAIn]:DELay:TIME <NR3>`  
`HORizontal[:MAIn]:DELay:TIME?`

**Related Commands** [HORizontal\[:MAIn\]:DELay:MODE](#), [HORizontal\[:MAIn\]:DELay:POSition](#)

**Arguments** <NR3> specifies the time base trigger delay time setting, typically represented in seconds.

**Examples** `HORIZONTAL:DELAY:TIME 5.0E-3` sets the time base trigger delay time to 5 ms.  
`HORIZONTAL:DELAY:TIME?` might return `:HORIZONTAL:MAIN:DELAY 5.0000E-05`, indicating that the time delay setting is 5  $\mu$ s.

## HORizontal:MAIn:INTERPRatio? (Query Only)

This query-only command returns the Horizontal Main time base interpolation ratio. The ratio is useful for calculating the maximum sampling rate when users request sampling rates that are faster than the maximum rate specified by the acquisition hardware.

The default maximum sampling rate as specified by the acquisition hardware is 1.

An interpolation ratio other than 1 means that displayed and retrieved waveforms contain additional points that are derived through interpolation. At such times, the retrieved waveform data is actually 16-bit data. You should set the data width accordingly.

**Group** Horizontal

**Syntax** `HORIZONTAL:MAIN:INTERPRATIO?`

**Examples** `HORIZONTAL:MAIN:INTERPRATIO?` might return `:HORIZONTAL:MAIN:INTERPRATIO 10.0`.

## HORizontal[:MAIn]:POSition

**Group** Horizontal

**Syntax** `HORIZONTAL[:MAIN]:POSITION <NR3>`  
`HORIZONTAL[:MAIN]:POSITION?`

**Related Commands** [HORizontal\[:MAIn\]:DELay:MODE](#)

**Arguments** <NR3> argument can range from 0 to  $\approx$ 100 and is the percentage of the waveform that is displayed left of the center graticule.

**NOTE.** *The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit = 100 - 1/record length).*

**Examples**      `HORIZONTAL:MAIN:POSITION 5.000E+01` sets the horizontal position of the waveform such that 10 percent of the waveform is to the left of center of the screen.

`HORIZONTAL:MAIN:POSITION?` might return `:HORIZONTAL:MAIN:POSITION 5.0000E+01`, indicating that the horizontal position of the waveform on the screen is currently set to 50 percent.

## HORizontal:MAIn:UNIts

This command sets or queries the units for the horizontal time base. It is equivalent to setting the `HORizontal:MAIn:UNIts:STRing`.

**Group**      Horizontal

**Syntax**      `HORizontal:MAIn:UNIts <QString>`  
`HORizontal:MAIn:UNIts?`

**Related Commands**      [HORizontal:MAIn:UNIts:STRing](#)

**Arguments**      `<QString>` is the time base units string.

**Examples**      `HORIZONTAL:MAIN:UNITS"LUM"` sets the time base multiplier to 5.5, the units to lumens and the time base units offset to 2.9.

`HORIZONTAL:MAIN:UNITS?` might return `:HORIZONTAL:MAIN:UNITS:STRING "lum"`.

## HORizontal:MAIn:UNIts:STRing

This command sets or queries the units string for the horizontal time base.

**Group**      Horizontal

**Syntax**      `HORizontal:MAIn:UNIts:STRing <QString>`  
`HORizontal:MAIn:UNIts:STRing?`



**Related Commands** [HORizontal:MAIn:UNIts](#)

**Arguments** <QString> is the time base units string.

**Examples** `HORIZONTAL:MAIN:UNITS:STRING"LUM"` sets the units string for the time base trigger delay to lumens.

`HORIZONTAL:MAIN:UNITS:STRING?` might return `:HORIZONTAL:MAIN:UNITS:STRING "lum"`, indicating that the units string for the time base trigger delay is set to lumens.

## HORizontal:MODE

This command set or queries the horizontal mode. Auto mode is the factory default.

**Group** Horizontal

**Syntax** `HORizontal:MODE {AUTO|CONSTant|MANuaL}`  
`HORizontal:MODE?`

**Related Commands** [HORizontal:MODE:AUTO:LIMITrecordlen](#), [HORizontal:MODE:RECOrdlength](#), [HORizontal:MODE:SAMPLERate](#), [HORizontal:MODE:SCAle](#)

**Arguments** `AUTO` selects the automatic horizontal model. Auto mode attempts to keep record length constant as you change the time per division setting. Record length is read only.

`CONSTANT` selects the constant horizontal model. Constant mode attempts to keep sample rate constant as you change the time per division setting. Record length is read only.

`MANUAL` selects the manual horizontal model. Manual mode lets you change sample mode and record length. Time per division or Horizontal scale is read only.

**Examples** `HORIZONTAL:MODEAUTO` sets the horizontal mode to auto.

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

## HORizontal:MODE:AUTO:LIMITrecordlen

This command sets or queries the record length limit used by the auto horizontal mode.

**Group** Horizontal

**Syntax** `HORizontal:MODE:AUTO:LIMITrecordlen <NR1>`  
`HORizontal:MODE:AUTO:LIMITrecordlen?`

**Arguments** <NR1> is the record length limit in samples.

**Examples** `HORIZONTAL:MODE:AUTO:LIMITRECORDLEN1000` sets the record length limit to 1000 samples.

`HORIZONTAL:MODE:AUTO:LIMITRECORDLEN?` might return  
`:HORIZONTAL:MODE:AUTO:LIMITRECORDLEN 1000000`, indicating that the record length limit is set to 1,000,000 samples.

## 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 Auto and Constant mode.

**Examples** `HORIZONTAL:MODE:RECORDLENGTH1000` 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.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal:MODE:SAMPLERate &lt;NR1&gt;</code> <code>HORizontal:MODE:SAMPLERate?</code>
<b>Arguments</b>	<NR1> is the sample rate in samples per second.
<b>Examples</b>	<code>HORIZONTAL:MODE:SAMPLERATE 1e6</code> sets the sample rate to 1 million samples per second.  <code>HORIZONTAL:MODE:SAMPLERATE?</code> might return <code>:HORIZONTAL:MODE:SAMPLERATE 5.0000E+6</code> , indicating that the sample rate is set to 5 million samples per second.

## HORizontal:MODE:SCALE

This command sets or queries the horizontal scale.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal:MODE:SCALE &lt;NR1&gt;</code> <code>HORizontal:MODE:SCALE?</code>
<b>Arguments</b>	<NR1> is the horizontal scale in seconds per division. The horizontal scale is read only for Manual mode.
<b>Examples</b>	<code>HORIZONTAL:MODE:SCALE2e-9</code> sets the horizontal scale to 2 ns per division.  <code>HORIZONTAL:MODE:SCALE?</code> might return <code>:HORIZONTAL:MODE:SCALE 20.0000E-6</code> , indicating that the horizontal scale is set to 10 $\mu$ s per division.

## HORizontal:POSition

This command either positions the waveform horizontally on the display or returns its horizontal position on the display. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel.

<b>Group</b>	Horizontal
--------------	------------

**Syntax**      `HORizontal:POSITION <NR3>`  
`HORizontal:POSITION?`

**Related Commands**    [HORizontal\[:MAIn\]:DELay:MODE](#)

**Arguments**      <NR3> is from 0 to ≈100 and is the percentage of the waveform that is displayed left of the center graticule.

---

**NOTE.** *The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit = 100 - 1/record length).*

---

**Examples**      `HORIZONTAL:POSITION 10` sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

`HORIZONTAL:POSITION?` might return `:HORIZONTAL:POSITION 5.0000E+01` indicates the horizontal position of the waveform on the screen.

## HORizontal:ROLL

This command sets or queries the Roll Mode status. Use Roll Mode when you want to view data at very slow sweep speeds. It is useful for observing data samples on the screen as they occur. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu, selecting the Acquisition tab, and setting the Roll Mode to Auto or Off.

**Group**          Horizontal

**Syntax**      `HORizontal:ROLL {AUTO|OFF|ON}`  
`HORizontal:ROLL?`

**Arguments**      `AUTO` enables Roll Mode, if the time/division is set appropriately.  
`OFF` disables Roll Mode.  
`ON` enables Roll Mode, if the time/division is set appropriately.

**Examples**      `HORIZONTAL:ROLLON` enables Roll Mode.  
`HORIZONTAL:ROLL?` might return `:HORIZONTAL:ROLL OFF`, indicating that the Roll Mode is disabled.

## HORizontal:TIMEStamp:CH<x>? (Query Only)

This query returns the horizontal timebase for the channel. The channel is specified by x. The value of x can range from 1 through 4.

**Group** Horizontal

**Syntax** HORizontal:TIMEStamp:CH<x>?

**Examples** HORizontal:TIMEStamp:CH1? might return  
:HORizontal:TIMEStamp:CH1 "timestamp not available", indicating that the timebase for the Channel 1 is not available.

## HORizontal:TIMEStamp:REF<x>? (Query Only)

This query returns the horizontal timebase for the reference waveform. The reference is specified by x. The value of x can range from 1 through 4.

**Group** Horizontal

**Syntax** HORizontal:TIMEStamp:REF<x>?

**Examples** HORizontal:TIMEStamp:REF1? might return  
:HORizontal:TIMEStamp:REF1 "timestamp not available", indicating that the timebase for the Reference 1 waveform is not available.

## ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware.

**Group** Miscellaneous

**Syntax** ID?

**Related Commands** [\\*IDN?](#)

**Examples** ID? might return :TEK/DPO7104,CF:91.1CT,FV:4.0.0, indicating that the instrument model number is set to TEK/DPO7104, configured format is set to CF:91.1CT, and firmware version number is set to FV:4.0.0 .

## \*IDN? (Query Only)

This query-only command returns the instrument identification code.

**Group** Miscellaneous

**Syntax** \*IDN?

**Related Commands** [ID?](#)

**Examples** \*IDN? might return :TEKTRONIX,DPO7104,10000001,CF:91.1CT FV:01.00.912,, indicating the instrument model number, serial number, configured number, and firmware version number.

## LIMit? (Query Only)

This query-only command returns all settings for the Limit commands.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit?

**Examples** LIMIT? might return :LIMIT:BEEP 0;COMPARE:CH1 NONE;CH2 NONE;CH3 NONE;CH4 NONE;REF1 NONE;REF2 NONE;REF3 NONE;REF4 NONE;MATH1 NONE;MATH2 NONE;MATH3 NONE;MATH4 NONE;;LIMIT:HARDCOPY 0;STATE 0;TEMPLATE:DESTINATION REF1;SOURCE CH2;TOLERANCE:HORIZONTAL 40.0000E-3;VERTICAL 40.0000E-3;;LIMIT:SRQ 0;STOPONVIOLATION 0;SAVEWFM 0;SAVEWFM:FILENAME "C:/TekScope/waveforms/";:LIMIT:LOG 0;EMAIL 0;HIGHLIGHTHITS 1;LOCK 1.

## LIMit:BEEP

This command causes the instrument to beep when the waveform data exceeds the limits set in the limit test (LIMit:STATE must be on).

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:BEEP {OFF ON <NR1>} LIMit:BEEP?
<b>Related Commands</b>	<a href="#">LIMit:STATE</a>
<b>Arguments</b>	OFF disables the beep. ON enables the beep. <NR1> = 0 disables the beep; any other value enables the beep.
<b>Examples</b>	LIMIT:BEEPON causes a beep when any waveform data exceeds the limits specified in the limit test.  LIMIT:BEEP? might return :LIMIT:BEEP 0, indicating that the instrument will not beep when waveform data exceeds the limits specified in the limit test.

## LIMit:COMpare (No Query Form)

This command resets the Limit Test comparison template.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:COMpare {RESet}
<b>Arguments</b>	RESet resets the Limit template to “None.”
<b>Examples</b>	LIMIT:COMPARE RESET sets the Limit template to None.

## LIMit:COMpare:CH<x>

This command sets or queries the template against which to compare the waveform acquired from the channel specified by x. The template can be a waveform saved in any of the reference locations (REF1 through REF4) or none.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:COMpare:CH<x> {NONE REF<x>} LIMit:COMpare:CH<x>?
<b>Arguments</b>	NONE turns off template testing for the channel specified by CH<x>.  REF<x> selects which channel waveform to use as the template against which to compare the waveforms acquired by the specified channel (CH<x>).
<b>Examples</b>	LIMIT:COMPARE:CH1 REF1 specifies REF1 as the template waveform against which to compare waveforms acquired using Channel 1.  LIMIT:COMPARE:CH2? might return :LIMIT:COMPARE:CH2 REF4, indicating that the waveform acquired by Channel 2 is compared to the template waveform stored in REF4.

## LIMit:COMpare:MATH<x>

This command sets or queries the template against which to compare the math waveform specified by x, which can be 1 through 4. The template can be a waveform saved in any of the reference locations (REF1 through REF4) or none.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:COMpare:MATH<x> {NONE REF<x>} LIMit:COMpare:MATH<x>?
<b>Arguments</b>	NONE argument turns off template testing for the math waveform specified by MATH<x>.



REF<x> argument selects which reference waveform to use as the template against which to compare the math waveform specified by MATH<x>.

**Examples**    `LIMIT:COMPARE:MATH1 REF1` specifies REF1 as the template waveform against which to compare Math 1 waveform.

`LIMIT:COMPARE:MATH2?` might return `:LIMIT:COMPARE:MATH2 REF4`, indicating that the Math 2 waveform is compared to the template waveform stored in REF4.

## LIMit:COMpare:REF<x>

This command sets or queries the template against which to compare the reference waveform specified by x, which can be 1 through 4. The template can be a waveform saved in any of the reference locations (REF1 through REF4) or none.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group**    Limit Test

**Syntax**    `LIMit:COMpare:REF<x> {NONE|REF<x>}`  
`LIMit:COMpare:REF<x>?`

**Arguments**    NONE argument turns off template testing for the reference waveform specified by REF<x>.

REF<x> argument selects which reference waveform to use as the template against which to compare the reference waveform specified by REF<x>.

**Examples**    `LIMIT:COMPARE:REF2 REF1` specifies REF2 as the template waveform against which to compare Reference 1.

`LIMIT:COMPARE:REF2?` might return `:LIMIT:COMPARE:REF2 REF4`, indicating that the Reference 4 waveform is compared to the template waveform stored in REF2.

## LIMit:EMail

This command sets or queries whether an e-mail is generated when the source waveform data exceeds the limits specified for the limit test.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:EMail {OFF ON <NR1>} LIMit:EMail?
<b>Arguments</b>	<p>OFF disables generating an e-mail when the source waveform exceeds the test limits.</p> <p>ON enables generating an e-mail when the source waveform exceeds the test limits.</p> <p>&lt;NR1&gt; = 0 disables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is not generated; any other value enables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is generated. For queries, a 0 is returned if the e-mail feature is off; a 1 is returned if the e-mail feature is on.</p>
<b>Examples</b>	<p>LIMIT:EMAIL1 enables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is generated.</p> <p>LIMIT:EMAIL? might return :LIMIT:EMAIL 0, indicating that the e-mail feature is disabled and that an e-mail will not be generated when the source waveform exceeds the limits of the limit test.</p>

## LIMit:HARDCopy

This command sets or queries whether a hard copy operation is executed on the waveform when any waveform data exceeds the limit set in the limit test. LIMit:STATE must be set to ON for the hard copy operation to execute. The hard copy operation uses the port and prints in the format and layout specified by the HARDCopy commands.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:HARDCopy {OFF ON <NR1>} LIMit:HARDCopy?
<b>Related Commands</b>	<a href="#">HARDCopy:FILENAME</a> , <a href="#">HARDCopy:PORT</a> , <a href="#">LIMit:STATE</a>

**Arguments** OFF disables the hard copy operation.

ON turns on the hard copy operation for the waveform when any waveform data exceeds the limits set by the limit test.

<NR1> = 0 disables the hard copy operation; any other value turns on the hard copy operation for the waveform when any waveform data exceeds the limits set by the limit test. For queries, a 0 is returned if the hard copy operation is off; a 1 is returned if the hard copy operation is on.

**Examples** LIMIT:HARDCOPYON specifies that the hard copy operation will occur for the waveform when any waveform data exceeds the limits specified in the limit test.

LIMIT:HARDCOPY? might return :LIMIT:HARDCOPY 0, indicating that the hardcopy operation is disabled.

## LIMit:HIGHLIGHTHits

This command sets or queries whether violation highlighting occurs when limit testing is active, and, if the RESET argument is set, clears the highlighting.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit:HIGHLIGHTHits {OFF|ON|<NR1>}  
LIMit:HIGHLIGHTHits?

**Arguments** OFF disables violation highlighting when limit testing is active.

ON enables violation highlighting when limit testing is active.

<NR1> = 0 disables the violation highlighting when limit testing is active; any other value enables the violation highlighting feature when limit testing is active. For queries, a 0 is returned if the violation highlighting feature is off; a 1 is returned if the violation highlighting feature is on.

**Examples** LIMIT:HIGHLIGHTHITS1 enables the violation highlighting when the limit testing is active.

LIMIT:HIGHLIGHTHITS? might return :LIMIT:HIGHLIGHTHITS 0, indicating that the violation highlighting feature is disabled.

## LIMit:HIGHLIGHTHits:RESet (No Query Form)

This command resets the hits highlighting for limit testing.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit:HIGHLIGHTHits:RESet

**Examples** LIMIT:HIGHLIGHTHITS:RESET resets the violation highlighting displayed while limit testing.

## LIMit:LOCK

This command sets or queries whether vertical scaling and positioning affect both source and template for template comparison pairs.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit:LOCK {OFF|ON|<NR1>}  
LIMit:LOCK?

**Arguments** OFF disables the lock feature.

ON enables the lock feature.

<NR1> = 0 disables the lock feature; any other value enables the lock feature. For queries, a 0 is returned if the lock feature is off; a 1 is returned if the lock feature is on.

**Examples** LIMIT:LOCKON enables the lock feature so that vertical scaling and positioning affect both the source and the template for template comparison pairs.

LIMIT:LOCK? might return :LIMIT:LOCK 0, indicating that the lock feature is disabled.

## LIMit:LOG

This command sets or queries whether a log file is saved when the source waveform data exceeds the test limits.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:LOG {OFF ON <NR1>} LIMit:LOG?
<b>Arguments</b>	<p>OFF disables saving a log file when the source waveform exceeds the test limits.</p> <p>ON enables saving a log file when the source waveform exceeds the test limits.</p> <p>&lt;NR1&gt; = 0 disables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is not saved; any other value enables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is saved. For queries, a 0 is returned if the save log file feature is off; a 1 is returned if the save log file feature is on.</p>
<b>Examples</b>	<p>LIMIT:LOG1 enables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is saved.</p> <p>LIMIT:LOG? might return :LIMIT:LOG 0, indicating that the save log file feature is disabled and that a log file will not be saved when the source waveform exceeds the limits of the limit test.</p>

## LIMit:SAVEWFM

This command sets or queries whether the source waveform is saved when the source waveform data exceeds the test limits.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:SAVEWFM {OFF ON <NR1>} LIMit:SAVEWFM?

- Arguments**    OFF disables saving the source waveform when it exceeds the test limits.
- ON enables saving the source waveform when it exceeds the test limits.
- <NR1> = 0 disables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is not saved; any other value enables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is saved. For queries, a 0 is returned if the save waveform feature is off; a 1 is returned if the save waveform feature is on.
- Examples**     LIMIT:SAVEWFM1 enables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is saved.
- LIMIT:SAVEWFM? might return :LIMIT:SAVEWFM 0, indicating that the save waveform feature is disabled and that the source waveform will not be saved when it exceeds the limits of the limit test.

## LIMit:SAVEWFM:FILEName

This command sets or queries the path where waveforms or log files will be saved when the waveform data exceeds the limits set by the limit test.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

- Group**        Limit Test
- Syntax**        LIMit:SAVEWFM:FILEName <QString>  
                   LIMit:SAVEWFM:FILEName?
- Arguments**    <QString> argument is a string containing the path of where the waveform will be saved.
- Examples**     LIMIT:SAVEWFM:FILENAME"C:/TekScope/waveforms" sets the path where a waveform will be saved when it exceeds the limits of the limit test.
- LIMIT:SAVEWFM:FILENAME? might return :LIMIT:SAVEWFM:FILENAME "C:/TekScope/waveforms", indicating the path where the waveform will be saved when it exceeds the limits of the limit test.

## LIMit:SRQ

This command sets or queries whether a Service Request Interrupt (SRQ) is generated when the waveform data falls outside of the test limits.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:SRQ {OFF ON <NR1>} LIMit:SRQ?
<b>Arguments</b>	<p>OFF disables generation of an SRQ when the waveform data falls outside of the limits set by the limit test.</p> <p>ON enables generation of an SRQ when the waveform data falls outside of the limits set by the limit test. If an SRQ is generated, the instrument sends the status event "Limit testing failed."</p> <p>&lt;NR1&gt; = 0 disables sending an SRQ when the waveform data falls outside of the limits set by the limit test; any other value enables generation of an SRQ when the waveform data falls outside of the limits set by the limit test. For queries, a 0 is returned if SRQ is off; a 1 is returned if the SRQ is on.</p>
<b>Examples</b>	<p>LIMIT:SRQ1 specifies that an SRQ will be generated when the waveform data falls outside of the limits set by the limit test. If an SRQ is generated, an status event of "Limit testing failed" is output.</p> <p>LIMIT:SRQ? might return :LIMIT:SRQ 0, indicating that an SRQ will not be generated when the waveform data falls outside of the limits set by the limit test.</p>

## LIMit:STATE

This command sets limit testing on or off or queries whether limit testing is in effect.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:STATE {OFF ON <NR1>} LIMit:STATE?

**Arguments** OFF disables limit testing.  
ON turns on limit testing of waveforms.  
<NR1> = 0 disables the limit testing operation; any other value turns on limit testing of waveforms. For queries, a 0 is returned if limit testing is off; a 1 is returned if the limit testing is on.

**Examples** LIMIT:STATEON specifies that limit testing of waveforms is on.  
LIMIT:STATE? might return :LIMIT:STATE 0, indicating that the limit testing of waveforms is off.

## LIMit:STATus? (Query Only)

This command queries the state of limit testing.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit:STATus?

**Examples** LIMIT:STATUS? might return :LIMIT:STATUS PASS, indicating that the limit test is passing.

## LIMit:STOPOnviolation

This command sets or queries whether acquisitions are stopped when the waveform data exceeds the test limits.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** LIMit:STOPOnviolation {OFF|ON|<NR1>}  
LIMit:STOPOnviolation?



<b>Arguments</b>	<p>OFF disables the stop on violation feature.</p> <p>ON enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped.</p> <p>&lt;NR1&gt; = 0 disables the stop on violation feature; any other value enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped. For queries, a 0 is returned if the stop on violation feature is off; a 1 is returned if the stop on violation feature is on.</p>
<b>Examples</b>	<p>LIMIT:STOPONVIOLATION1 enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped.</p> <p>LIMIT:STOPONVIOLATION? might return :LIMIT:STOPONVIOLATION 0, indicating that the stop on violation feature is disabled.</p>

## LIMit:TEMPLate:DESTination

This command sets or queries the destination reference waveform that the [LIMit:TEMPLate:STORe](#) command will use.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	<pre>LIMit:TEMPLate:DESTination {REF1 REF2 REF3 REF4} LIMit:TEMPLate:DESTination?</pre>
<b>Arguments</b>	REF<x> specifies the reference waveform destination in which the template waveform is to be stored.
<b>Examples</b>	<p>LIMIT:TEMPLATE:DESTINATIONREF2 specifies that the template waveform referred to with the <a href="#">LIMit:TEMPLate:STORe</a> command is stored as the REF2 waveform.</p> <p>LIMIT:TEMPLATE:DESTINATION? might return :LIMIT:TEMPLATE:DESTINATION REF3, indicating the REF3 waveform is used to store the template waveform referred to with the <a href="#">LIMit:TEMPLate:STORe</a> command.</p>

## LIMit:TEMPLate:SOURce

This command sets or queries the channel, math waveform, or reference waveform that the [LIMit:TEMPLate:STORe](#) command will use.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:TEMPLate:SOURCE {CH1 CH2 CH3 CH4 MATH1 MATH2 MATH3 MATH4 REF1 REF2 REF3 REF4} LIMit:TEMPLate:SOURCE?
<b>Arguments</b>	CH1–CH4 specifies the channel used by the LIMit:TEMPLate:STORe command.  MATH1–MATH4 specifies the math waveform used by the LIMit:TEMPLate:STORe command.  REF1–REF4 specifies the reference waveform used by the LIMit:TEMPLate:STORe command.
<b>Examples</b>	LIMIT:TEMPLATE:SOURCECH2 specifies that the template waveform for limit tests is the waveform currently acquired using Channel 2.  LIMIT:TEMPLATE:SOURCE? might return :LIMIT:TEMPLATE:SOURCE MATH3, indicating that the template waveform for limit tests is the waveform currently stored as the Math 3 waveform.

## LIMit:TEMPLate:STORe (No Query Form)

This command (no query form) saves the specified source waveform to the specified reference or file name.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

<b>Group</b>	Limit Test
<b>Syntax</b>	LIMit:TEMPLate:STORe <wfm>,{REF<x> <file path>}
<b>Arguments</b>	<wfm> specifies the waveform that will be saved as the template. The source of the waveform can be CH<x> (where x is 1 through 4 for four channel instruments), MATH<x>, or Ref<x> (where x is 1 through 4).  REF<x> specifies a reference location in which the template waveform will be stored. The reference waveform specified by x, which can be 1 through 4.

<file path> specifies a file path where the template waveform will be stored. This argument is a quoted string.

- Examples**    `LIMIT:TEMPLATE:STORECH3,REF1` creates a template reference waveform, REF1, using Channel 3 as its source.
- `LIMIT:TEMPLATE:STORECH1,"TEMPLATE.WFM"` creates a template reference waveform, template.wfm, using Channel 1 as the source.

## LIMit:TEMPLate:TOLerance:HORIZontal

This command sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

- Group**        Limit Test
- Syntax**        `LIMit:TEMPLate:TOLerance:HORIZontal <NR3>`  
                   `LIMit:TEMPLate:TOLerance:HORIZontal?`
- Arguments**    <NR3> is the amount, in horizontal divisions, by which the current source waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.
- Examples**        `LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL1.0` specifies that the current waveform is close enough to the template waveform if it is within  $\pm 1.0$  horizontal division.
- `LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL?` might return  
                   `:LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL 2.0`, indicating that the current waveform is close enough to the template waveform if it is within  $\pm 2.0$  horizontal divisions.

## LIMit:TEMPLate:TOLerance:VERTical

This command sets or queries the amount, in units of vertical divisions, by which the source waveform is varied vertically when creating the destination waveform.

---

**NOTE.** *The command works only if the LT Limit Test option is enabled.*

---

**Group** Limit Test

**Syntax** `LIMit:TEMPLate:TOLerance:VERTical <NR3>`  
`LIMit:TEMPLate:TOLerance:VERTical?`

**Arguments** <NR3> is the amount, in vertical divisions, by which the current source waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.

**Examples** `LIMIT:TEMPLATE:TOLERANCE:VERTICAL1.0` specifies that the current waveform is close enough to the template waveform if it is within  $\pm 1.0$  vertical division.

`LIMIT:TEMPLATE:TOLERANCE:VERTICAL?` might return  
`:LIMIT:TEMPLATE:TOLERANCE:VERTICAL 2.0`, indicating that the current waveform is close enough to the template waveform if it is within  $\pm 2.0$  vertical divisions.

## LOCK

This command enables or disables all front panel buttons and knobs, including the touch screen. There is no front panel equivalent.

**Group** Miscellaneous

**Syntax** `LOCK {ALL|NONE}`  
`LOCK?`

**Related Commands** [UNLock](#)

**Arguments** `ALL` disables all front panel controls.

`NONE` enables all front panel controls. This is equivalent to the `UNLock ALL` command.

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** LOCKALL locks the front panel controls.

LOCK? might return :LOCK NONE, indicating that the front panel controls are enabled by this command.

## \*LRN? (Query Only)

This query-only command returns the commands that list the instrument settings (except for configuration information for the calibration values), 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: :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT;: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;:VARIABLE:EVENT:REPORT BOTH;:AUXOUT:SOURCE ATRIGGER;EDGE FALLING;:CMDBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW TEMPERATURE;RECORDVIEW NORMAL;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.

## MARK

This commands moves to the next or previous reference mark, or it queries the current mark setup.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK {NEXT PREVIOUS} MARK?
<b>Arguments</b>	NEXT moves to the next reference mark on the right. PREVIOUS moves to the next reference mark on the left.
<b>Examples</b>	MARK NEXT moves to the next reference mark on the right.  MARK? might return :MARK:SELECTED:STATE 1;START 64.4000;FOCUS 64.4000;ZOOM:POSITION 64.4000;SCALE -3.1416;:MARK:SELECTED:END 64.4000;SOURCE CH1;OWNER SEARCH1;MARKSINCOLUMN 1;LABEL "+Edge: Level=420.32mV";:MARK:FREE 1011;TOTAL 13, which is the current mark setup.

## MARK:CREATE (No Query Form)

This command creates a mark on a specified waveform or all waveforms in a column.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:CREATE {CH<x> MATH<x> REF<x> COLUMN}
<b>Arguments</b>	CH<x> creates the mark on a channel waveform, where <x> is the channel number and can be 1, 2, 3, or 4.  MATH<x> creates the mark on the math waveform, where <x> is the channel number and can be 1, 2, 3, or 4.  REF<x> creates the mark on a reference waveform, where <x> is the reference waveform number can be 1, 2, 3, or 4.  COLUMN creates marks on all waveforms in the current zoom pixel column.
<b>Examples</b>	MARK:CREATECH1 creates a mark on Channel 1.

## MARK:DELEte (No Query Form)

This command deletes a mark on a particular waveform, all waveforms in a column, the selected mark, or all marks.

**Group** Search and Mark

**Syntax** MARK:DELEte {CH<x> | MATH<x> | SELECTED | REF<x> | ALL | COLUMN}

**Arguments** CH<x> deletes the mark on a channel waveform, where <x> is the channel number and can be 1, 2, 3, or 4.

MATH<x> deletes the mark on the math waveform, where <x> is the channel number and can be 1, 2, 3, or 4.

SELECTED deletes the mark on the selected waveform.

REF<x> deletes the mark on a reference waveform, where <x> is the reference waveform number and can be 1, 2, 3, or 4.

ALL deletes all marks on all waveforms.

COLUMN deletes marks on all waveforms in the current zoom pixel column.

**Examples** MARK:DELETECH1 deletes the mark on the Channel 1 waveform.

## MARK:FREE? (Query Only)

This query-only command returns how many marks are available for use.

**Group** Search and Mark

**Syntax** MARK:FREE?

**Examples** MARK:FREE? might return :MARK:FREE 1011, indicating there are 1,011 marks available for use.

## MARK:SELECTED:END? (Query Only)

This query-only command returns the end of the selected mark, 0 to 100% of the waveform.

**Group** Search and Mark

**Syntax** MARK:SELECTED:END?

**Examples** MARK:SELECTED:END? might return :MARK:SELECTED:END 93.0000, indicating that the selected mark is at 93% of the waveform.

## MARK:SELECTED:FOCUS? (Query Only)

This query-only command returns the focus of the selected mark, 0 to 100% of the waveform.

**Group** Search and Mark

**Syntax** MARK:SELECTED:FOCUS?

**Examples** MARK:SELECTED:FOCUS? might return :MARK:SELECTED:FOCUS 50.0000, indicating that the selected mark is at 50% of the waveform.

## MARK:SELECTED:LABel

This command sets or queries the label for the selected mark.

**Group** Search and Mark

**Syntax** MARK:SELECTED:LABel <QString>  
MARK:SELECTED:LABel?

**Arguments** <QString> is the quoted string label for the mark.

**Examples** MARK:SELECTED:LABEL "Test 42.3" sets the selected mark to a User mark and changes the mark description to Test 42.3.

MARK:SELECTED:LABEL? might return :MARK:SELECTED:LABEL "+Edge: level=420.32mV", indicating that the label description for the selected mark is +Edge: level=420.32mV.



## MARK:SELECTED:MARKSINCOLUMN? (Query Only)

This query-only command returns the number of marks in the current zoom pixel column.

**Group** Search and Mark

**Syntax** MARK:SELECTED:MARKSINCOLUMN?

**Arguments** MARK:SELECTED:MARKSINCOLUMN? might return :MARK:SELECTED:MARKSINCOLUMN 1, indicating there is 1 mark in the pixel column.

## MARK:SELECTED:OWNER? (Query Only)

This query-only command returns the owner of the selected mark.

**Group** Search and Mark

**Syntax** MARK:SELECTED:OWNER?

**Examples** MARK:SELECTED:OWNER? might return: :MARK:SELECTED:OWNER SEARCH1, indicating that the selected mark is a Search1 mark.

## MARK:SELECTED:SOURCE? (Query Only)

This query-only command returns the source waveform for the selected mark.

**Group** Search and Mark

**Syntax** MARK:SELECTED:SOURCE?

**Examples** MARK:SELECTED:SOURCE? might return :MARK:SELECTED:SOURCE CH2, indicating that Channel 2 is the selected mark source.

## MARK:SELECTED:START? (Query Only)

This query-only command returns the starting point of the selected mark, 0 to 100% of the waveform.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:SELECTED:START?
<b>Examples</b>	MARK:SELECTED:START? might return :MARK:SELECTED:START 5.8000, indicating that the selected mark start location is 5.8% of the waveform.

## MARK:SELECTED:STATE

This command sets or queries the on or off state of the selected mark. The selected mark is at or near the center of the display. If you press the Set/Clear button, this mark will disappear.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:SELECTED:STATE {<NR1> OFF ON} MARK:SELECTED:STATE?
<b>Arguments</b>	OFF argument disables selected mark.  ON argument enables the selected mark.  <NR1>= 0 disables the selected mark; any other value enables the selected mark. For queries, a 0 is returned if the selected mark state is off; a 1 is returned if the selected mark state is on.
<b>Examples</b>	MARK:SELECTED:STATEON turns the selected mark state on.  MARK:SELECTED:STATE? might return :MARK:SELECTED:STATE 0, indicating the selected mark state is off.

## MARK:SELECTED:ZOOM:POSITION? (Query Only)

This query-only command returns the position of the selected mark, of the zoom overview window. If the return value is less than zero then the zoom scale will not be changed when Next or Prev makes this the selected mark.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:SELECTED:ZOOM:POSITION?
<b>Examples</b>	MARK:SELECTED:ZOOM:POSITION? might return :MARK:SELECTED:ZOOM:POSITION 27.7000, indicating that the position is 27.7%.

## MARK:SELECTED:ZOOM:SCALE? (Query Only)

This query-only command returns the scale of the selected mark of the zoom overview window. The returned value might be  $< 0$  (for example from Search), which means that the zoom scale will not be changed when Next or Prev makes this the selected mark.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:SELECTED:ZOOM:SCALE?
<b>Examples</b>	MARK:SELECTED:ZOOM:SCALE? might return :MARK:SELECTED:ZOOM:SCALE 5.0, indicating that the zoom scale value 5.0 will be used when a mark is the selected mark.

## MARK:TOTAL? (Query Only)

This query-only command returns how many marks are currently in use.

<b>Group</b>	Search and Mark
<b>Syntax</b>	MARK:TOTAL?
<b>Examples</b>	MARK:TOTAL? might return :MARK:TOTAL 210, indicating that 210 marks are in use.

## MASK? (Query Only)

This query-only command returns the states of all settable mask parameters.

<b>Group</b>	Mask
<b>Syntax</b>	MASK?
<b>Examples</b>	MASK? might return the following response: :MASK:FILTER 0;AUTOADJUST 0;AUTOADJUST:HDELTA 10;VDELTA 10;:MASK:AUTOSET:AUTOADJUST 1;HPOS 1;HSCALE 1;MODE MANUAL;OFFSETADJ 1;TRIGGER 1;VPOS 1;VSCALE 1;STANDARD NONE;USER:ONE 0.0000;ZERO 0.0000;TYPE NORMALIZED;:MASK:SOURCE CH1;COUNT:FAILURES 0;HITS 0;SEG1:.

## MASK:AUTOAdjust

This command optimizes or queries the signal position within the mask to minimize hits. It sets a mode so that the MASK:SOURCE waveform is compared against the mask and is shifted up, down, left, or right to minimize the hits.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:AUTOAdjust {ON OFF <NR1>} MASK:AUTOAdjust?
<b>Arguments</b>	ON enables the autoadjust function. OFF disables the autoadjust function. <NR1> = 0 disables the autoadjust function; any other value enables it.
<b>Examples</b>	MASK:AUTOADJUSTON enables the autoadjust function. MASK:AUTOADJUST? might return :MASK:AUTOADJUST 1, indicating that the autoadjust function is enabled.

## MASK:AUTOAdjust:HDELTA

This command sets or returns how far autofit searches horizontally.

Autofit moves the waveform vertically and/or horizontally to reduce the number of hits within a given mask. If infinite or variable persistence is enabled, these movements will clear any persistent data. If Autofit makes frequent adjustments, there might be very little or even no persistent data displayed.

<b>Group</b>	Mask
--------------	------

<b>Syntax</b>	<code>MASK:AUTOAdjust:HDELTA {&lt;NR3&gt;}</code> <code>MASK:AUTOAdjust:HDELTA?</code>
<b>Arguments</b>	<NR3> is a floating point number that represents a percent of a division.
<b>Examples</b>	<code>MASK:AUTOADJUST:HDELTA0.6</code> sets the horizontal autofit search to 0.6 division. <code>MASK:AUTOADJUST:HDELTA?</code> might return <code>:MASK:AUTOADJUST:HDELTA 2</code> , indicating that the horizontal autofit search is set to 2 divisions.

## MASK:AUTOAdjust:VDELTA

This command sets or returns how far autofit searches vertically.

Autofit moves the waveform vertically and/or horizontally to reduce the number of hits within a given mask. If infinite or variable persistence is enabled, these movements will clear any persistent data. If Autofit makes frequent adjustments, there might be very little or even no persistent data displayed.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:AUTOAdjust:VDELTA {&lt;NR3&gt;}</code> <code>MASK:AUTOAdjust:VDELTA?</code>
<b>Arguments</b>	<NR3> is a floating point number that represents a percent of a division.
<b>Examples</b>	<code>MASK:AUTOADJUST:VDELTA0.6</code> sets the vertical autofit search to 0.6 of a division. <code>MASK:AUTOADJUST:VDELTA?</code> might return <code>:MASK:AUTOADJUST:VDELTA 2</code> , indicating that the vertical autofit search is set to 2 divisions.

## MASK:AUTOSet:AUTOAdjust

This command sets or returns whether the instrument settings are automatically matched to signal characteristics and specific mask requirements. It controls what happens at the end of a mask autoset. If ON, the horz/vert auto adjustment is run ONCE to potentially improve the result of mask autoset.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general instrument autoset function.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:AUTOSet:AUTOAdjust {ON OFF &lt;NR1&gt;}</code> <code>MASK:AUTOSet:AUTOAdjust?</code>
<b>Arguments</b>	<p>ON enables the autoset:autoadjust function.</p> <p>OFF disables the autoset:autoadjust function.</p> <p>&lt;NR1&gt; = 0 disables the autoset:autoadjust function; any other value enables it.</p>
<b>Examples</b>	<p><code>MASK:AUTOSET:AUTOADJUSTON</code> enables the autoset autoadjust function.</p> <p><code>MASK:AUTOSET:AUTOADJUST?</code> might return <code>:MASK:AUTOSET:AUTOADJUST 1</code>, indicating that the autoset:autoadjust function is enabled.</p>

## MASK:AUTOSet:HPOS

This command controls whether the autoset algorithm will attempt to change the horizontal position of the signal while attempting to do a mask autoset. The default is ON.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general instrument autoset function.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:AUTOSet:HPOS {ON OFF &lt;NR1&gt;}</code> <code>MASK:AUTOSet:HPOS?</code>
<b>Arguments</b>	<p>ON enables the autoset:hpos function.</p> <p>OFF disables the autoset:hpos function.</p> <p>&lt;NR1&gt; = 0 disables the autoset:hpos function; any other value disables enables it.</p>
<b>Examples</b>	<p><code>MASK:AUTOSET:HPOSON</code> enables the autoset:hpos function.</p> <p><code>MASK:AUTOSET:HPOS?</code> might return <code>:MASK:AUTOSET:HPOS 1</code>, indicating that the autoset:hpos function is enabled.</p>

## MASK:AUTOSet:HSCAlE

This command controls whether the autosegment algorithm will attempt to change the horizontal scale while attempting a mask autosegment.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general instrument autosegment function.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:AUTOSet:HSCAlE {ON OFF <NR1>} MASK:AUTOSet:HSCAlE?
<b>Arguments</b>	ON enables the autosegment:hscale function. OFF disables the autosegment:hscale function. <NR1> = 0 disables the autosegment:hscale function; any other value enables it.
<b>Examples</b>	MASK:AUTOSET:HSCALEON enables the autosegment:hscale function. MASK:AUTOSET:HSCALE? might return :MASK:AUTOSET:HSCALE 1, indicating that the autosegment:hscale function is enabled.

## MASK:AUTOSet:MODE

This command controls whether a mask autosegment will be automatically done after a standard mask is selected. The autosegment will never happen if the standard mask is selected from GPIB, since a "SET?" sent back to the oscilloscope should not perform a mask autosegment.

The default value is: manual.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general instrument autosegment function.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:AUTOSet:MODE {MANua1 AUTO} MASK:AUTOSet:MODE?
<b>Arguments</b>	MANua1 mask autosegment is set manually. AUTO mask autosegment is set automatically.

- Examples** MASK:AUTOSET:MODEAUTO sets the mask autosest manually.
- MASK:AUTOSET:MODE? might return :MASK:AUTOSET:MODE MANUAL, indicating that the mask autosest is set manually.

## MASK:AUTOSet:OFFSEtAdj

This command sets mask autosest not to enforce the rule that, for pulse standards, require 0 V to be in a certain place in the mask. Instead, mask autosest will attempt to measure the DC offset in the signal and use oscilloscope offset or position controls to compensate for the DC offset in the signal.

This command, like all the MASK:AUTOSET commands, affects only an autosest on a standard mask, not the general instrument autosest function.

**Group** Mask

**Syntax** MASK:AUTOSet:OFFSEtAdj {ON|OFF|<NR1>}  
MASK:AUTOSet:OFFSEtAdj?

**Arguments** ON enables the autosest:offsetadj function.  
OFF disables the autosest:offsetadj function.  
<NR1> = 0 disables the autosest:offsetadj function; any other value enables it.

**Examples** MASK:AUTOSET:OFFSEtADJON enables the autosest:offsetadj function.

MASK:AUTOSET:OFFSEtADJ? might return :MASK:AUTOSET:OFFSEtADJ 1, indicating that the autosest:offsetadj function is enabled.

## MASK:AUTOSet:STANdard

This command selects the standard mask in a mask autosest.

This command, like all the MASK:AUTOSET commands, affects only an autosest on a standard mask, not the general instrument autosest function.

**Conditions** This command is only valid if the instrument has Option MTM installed.

**Group** Mask



**Syntax** MASK:AUTOSet:STANdard  
 {ATATXG1|ATATXG2|ATATXG3|ATARXG1|ATARXG2  
 |ATARXG3|CLOCKCoax|CLOCKSymmetrical|D1|D2  
 |DS0Contra|DS0Double|DS0Single|DS0Timing|DS1  
 |DS1A|DS1C|DS2RATESymmetrical|DS2RATECoax  
 |DS2|DS3|DS4NA|DS4NA\_Max  
 |E1Symmetrical|E1Coax|E2|E3|E4\_0|E4\_1|ENET100FX  
 |ENET100STP |ENET100UTP |ENET1250|ENET1000BCX\_NTP2  
 |ENET1000BCX\_ATP2|ENET1000BCX\_ATP3|ENETXAU1\_Near  
 |ENETXAU1\_Far|FC133|FC266|FC531|FC1063|FC1063Draft  
 |FC2125|FC133E|FC266E|FC531E|FC1063E|FC1063E\_NBT  
 |FC1063E\_NDT|FC1063E\_NGT|FC1063E\_ABT|FC1063E\_ADT  
 |FC1063E\_AGT|FC1063E\_ABR|FC1063E\_ADR|FC1063E\_AGR  
 |FC2125E\_NBT|FC2125E\_NDT|FC2125E\_NGT|FC2125E\_ABT  
 |FC2125E\_ADT|FC2125E\_AGT|FC2125E\_ABR|FC2125E\_ADR  
 |FC2125E\_AGR|FC4250E\_ABR|FC4250E\_ABT|FC4250E\_ADR  
 |FC4250E\_ADT|FC4250E\_AGR|FC4250E\_AGT|FC4250E\_NBT  
 |FC4250E\_NDT|FC4250E\_NGT|FST1|FST2|FST3|FST4  
 |FST5|FST6|FW1394BS400BT1|FW1394BS400BT2  
 |FW1394BS800BT1|FW1394BS800BT2|FW1394BS1600BT1  
 |FW1394BS1600BT2|FW1394BS400B|FW1394BS800B  
 |FW1394BS1600B|G703DS1|G703DS3|HST1|HST2  
 |HST3|HST4|HST5|HST6|INF2\_5G  
 |INF2\_5GE|NONE|OC1|OC3|OC12|OC48|OC48\_FEC  
 |RATE32Mbit|RATE97Mbit|RIO\_DRV500M|RIO\_DRV750M  
 |RIO\_DRV1G|RIO\_DRV1\_5G|RIO\_DRV2G|RIO\_EDRV500M  
 |RIO\_EDRV750M |RIO\_EDRV1G|RIO\_EDRV1\_5G|RIO\_EDRV2G  
 |RIO\_RCV500M|RIO\_RCV750M|RIO\_RCV1G|RIO\_RCV1\_5G  
 |RIO\_RCV2G|RIO\_SERIAL\_1G|RIO\_SERIAL\_2G|RIO\_SERIAL\_3G  
 |SFI5\_XMITADATA2|SFI5\_XMITCDATA2|SFI5\_XMITACLK2  
 |SFI5\_XMITCCLK2|SFI5\_RCVBDATA2|SFI5\_RCVDDATA2  
 |SFI5\_RCVBCLK2|SFI5\_RCVDCLK2|SFI5\_XMITADATA3  
 |SFI5\_XMITCDATA3|SFI5\_XMITACLK3|SFI5\_XMITCCLK3  
 |SFI5\_RCVBDATA3|SFI5\_RCVDDATA3|SFI5\_RCVBCLK3  
 |SFI5\_RCVDCLK3|PCIEXPRESS\_Xmit|PCIEXPRESS\_Rcv  
 |SAS1\_5\_IR|SAS1\_5\_CR|SAS1\_5\_XR|SAS1\_5\_IR\_AASJ  
 |SAS1\_5\_CR\_AASJ|SAS1\_5\_XR\_AASJ|SAS1\_5\_SATA  
 |SAS3\_0\_IR|SAS3\_0\_CR|SAS3\_0\_XR|SAS3\_0\_IR\_AASJ  
 |SAS3\_0\_CR\_AASJ|SAS3\_0\_XR\_AASJ|SAS3\_0\_SATA  
 |STM0\_1|STM0\_0|STM0\_HDBX|STM1E\_0|STM1E\_1|STS1Pulse  
 |STS1Eye|STS3|STS3\_Max|TFI5\_2|TFI5\_3  
 |USERMask|VIDEO270|VIDEO292M|VIDEO360|VSROC192}  
 MASK:AUTOSet:STANdard?

**Arguments** ATATXG1 (Serial ATA, G1 Tx, 1.5 Gb/s)  
 ATATXG2 (Serial ATA, G2 Tx, 3.0 Gb/s)  
 ATATXG3 (Serial ATA, G3 Tx).  
 ATARXG1 (Serial ATA, G1 Rx 1.5 Gb/s)  
 ATARXG2 (Serial ATA, G2, Rx, 3.0 Gb/s)

ATARXG3 (Serial ATA, G3, Rx)  
G703D1 (ITU-T, G703 (10/98), DS1 Rate, 1.544 Mb/s)  
DS1 (ANSI T1.102-1993 (R1999), DS1, 1.544 Mb/s)  
DS1A (ANSI T1.102-1993 (R1999), DS1A, 2.048 Mb/s)  
DS1C (ANSI T1.102-1993 (R1999), DS1C, 3.152 Mb/s)  
DS2 (ANSI T1.102-1993 (R1999), DS2, 6.312 Mb/s)  
DS3 (ANSI T1.102-1993 (R1999), DS3, 44.736 Mb/s)  
DS4NA (ANSI T1.102-1993 (R1999), DS4NA, 139.26 Mb/s)  
DS4NA\_Max (ANSI T1.102-1993 (R1999), DSNA Max Output, 139.26 Mb/s)  
DS2RATECoax (ITU-T, G703 (10/98), D2 Rate Coax, 6.312 Mb/s)  
DS2RATESymmetrical (ITU-T, G703 (10/98), D2 Rate Sym, 6.312 Mb/s)  
E1Coax (ITU-T, G703 (10/98), E1 Coax Pair, 2.048 Mb/s)  
E1Symmetrical (ITU-T, G703 (10/98), E1 Sym Pair, 2.048 Mb/s)  
E2 (ITU-T, G703 (10/98), , 8.448 Mb/s)  
RATE32Mbit (ITU-T, G703 (10/98), 32.064 Mb/s)  
E3 (ITU-T, G703 (10/98), E3, 34.368 Mb/s)  
E4\_0 (ITU-T, G703 (10/98), E4 Binary 0)  
E4\_1 (ITU-T, G703 (10/98), E4 Binary 1)  
ENET100STP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, STP, 125 Mb/s )  
ENET100UTP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, UTP, 125 Mb/s)  
ENET1250 (IEEE Std 802.3 and ANSI X3.263-1995, GB Ethernet, 1.25 Gb/s)  
FC133 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 132.8 Mb/s)  
FC266 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 265.6 Mb/s)  
FC531 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 531.2 Mb/s)  
FC1063 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 1.065 Gb/s)  
FC1063Draft (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, Draft Rev 11)  
FC2125 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 2.125)  
FC133E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical 132.8 Mb/s)  
FC266E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 132.8 Mb/s)

FC531E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 531.2 Mb/s)

FC1063E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)

FC1063E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC1063E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC1063E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC1063E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC1063E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC1063E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC1063E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)

FC2125E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC2125E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC2125E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC2125E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC2125E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)

FC2125E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC2125E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC2125E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC2125E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)

ENET1000BCX\_NTP2 (1000B-CX Norm, TP2, 1.25 Gb/s)

ENET1000BCX\_ATP2 (1000B-CX Abs, TP2, 1.25 Gb/s)

ENET1000BCX\_ATP3 (1000B-CX Abs, TP3, 1.25 Gb/s)

ENETXAUI\_Near (10 Gigabit Attachment Unit Interface (XAUI), Near, 3.125 Gb/s)

ENETXAUI\_Far (10 Gigabit Attachment Unit Interface (XAUI), Far, 3.125 Gb/s)

FST1 (USB, FS:T1, 12 Mb/s)

FST2 (USB, FS:T2, 12 Mb/s)

FST3 (USB, FS:T3, 12 Mb/s)  
FST4 (USB, FS:T4: 12 Mb/s)  
FST5 (USB, FS:T5, 12 Mb/s)  
FST6 (USB, FS:T6, 12 Mb/s)  
FW1394BS400BT1 (IEEE 1394b, S400b T1, 491.5 Mb/s)  
FW1394BS400BT2 (IEEE 1394b, S400b T2, 491.5 Mb/s)  
FW1394BS800BT1 (IEEE 1394b, S800b T1, 983.0 Mb/s)  
FW1394BS800BT2 (IEEE 1394b, S800b T2, 983.0 Mb/s)  
FW1394BS1600BT1 (IEEE 1394b, S1600b T1, 1.966 Gb/s)  
FW1394BS1600BT2 (IEEE 1394b, S1600b T2, 1.966 Gb/s)  
FW1394BS400B (IEEE 1394b, S400 Optical, 491.5 Mb/s)  
FW1394BS800B (IEEE 1394b, S800 Optical, 988.0 Mb/s)  
FW1394BS1600B (IEEE 1394b, S1600 Optical, 1.966 Gb/s)  
G703DS3 (ITU-T, G703 (10/98))  
HST1 (USB, HS:T1, 480 Mb/s)  
HST2 (USB, HS:T2, 480 Mb/s)  
HST3 (USB, HS:T3, 480 Mb/s)  
HST4 (USB, HS:T4, 480 Mb/s)  
HST5 (USB, HS:T5, 480 Mb/s)  
HST6 (USB, HS:T6, 480 Mb/s)  
INF2\_5G (InfiniBand, IBTA Spec 1.0a, 2.5 Optical, 2.5 Gb/s)  
INF2\_5GE (InfiniBand, IBTA Spec 1.0a, 2.5 Electrical, 2.5 Gb/s)  
OC1 (GR 253-CORE Issue 3 9/21/2000 OC1/STM0, 51.84 Mb/s)  
OC3 (GR 253-CORE Issue 3 9/21/2000 OC1/STM1, 155.52, Mb/s)  
OC12 (GR 253-CORE Issue 3 9/21/2000 OC1/STM4, 622.08 Mb/s)  
OC48 (GR 253-CORE Issue 3 9/21/2000 OC1/STM16, 2.4883 Gb/s)  
OC48\_FEC (Forward Error Correction - CSA8000 mask, 2.666 Gb/s)  
PCIEXPRESS\_Xmit (PCI Express Transmitter, 2.5 Gb/s)  
PCIEXPRESS\_Rcv (PCI Express Receiver, 2.5 Gb/s)  
RATE97Mbit (ITU-T, G703 (10/98), 97 Mbit, 97.728 Mb/s)  
RIO\_DRV500M (Rapid IO Driver, 500 Mb/s)

RIO\_DRV750M (Rapid IO Driver, 750 Mb/s)  
 RIO\_DRV1G (Rapid IO Driver, 1 Gb/s)  
 RIO\_DRV1\_5G (Rapid IO Driver, 5 Gb/s)  
 RIO\_DRV2G (Rapid IO Driver, 2 Gb/s)  
 RIO\_EDRV500M (Rapid IO Extended Driver, 500 Mb/s)  
 RIO\_EDRV750M (Rapid IO Extended Driver, 750 Mb/s)  
 RIO\_EDRV1G (Rapid IO Extended Driver, 1 Gb/s)  
 RIO\_EDRV1\_5G (Rapid IO Extended Driver, 1.5 Gb/s)  
 RIO\_EDRV2G (Rapid IO Extended Driver, 2 Gb/s)  
 RIO\_RCV500M (Rapid IO Receiver, 500 Mb/s)  
 RIO\_RCV750M (Rapid IO Receiver, 750 Mb/s)  
 RIO\_RCV1G (Rapid IO Receiver, 1 Gb/s)  
 RIO\_RCV1\_5G (Rapid IO Receiver, 1.5 Gb/s)  
 RIO\_RCV2G (Rapid IO Receiver, 2 Gb/s)  
 RIO\_SERIAL\_1G (Rapid IO Serial, 1.25 Gb/s)  
 RIO\_SERIAL\_2G (Rapid IO Serial, 2.5 Gb/s)  
 RIO\_SERIAL\_3G (Rapid IO Serial, 3.25 Gb/s)  
 SFI5\_XMITADATA2 (SFI15 Transmit: Test Point A Data Signal 2, 2.488 Gb/s)  
 SFI5\_XMITCDATA2 (SFI15 Transmit: Test Point C Data Signal 2, 2.488 Gb/s)  
 SFI5\_XMITACLK2 (SFI15 Transmit: Test Point A Clock Signal 2, 2.488 Gb/s)  
 SFI5\_XMITCCLK2 (SFI15 Transmit: Test Point C Clock Signal 2, 2.488 Gb/s)  
 SFI5\_RCVBDATA2 (SFI15 Receive: Test Point B Data Signal 2, 2.488 Gb/s)  
 SFI5\_RCVDDATA2 (SFI15 Receive: Test Point D Data Signal 2, 2.488 Gb/s)  
 SFI5\_RCVBCLK2 (SFI15 Receive: Test Point B Clock Signal 2, 2.488 Gb/s)  
 SFI5\_RCVDCLK2 (SFI15 Receive: Test Point D Clock Signal 2, 2.488 Gb/s)  
 SFI5\_XMITADATA3 (SFI15 Transmit: Test Point A Data Signal 3, 3.125 Gb/s)  
 SFI5\_XMITCDATA3 (SFI15 Transmit: Test Point C Data Signal 3, 3.125 Gb/s)  
 SFI5\_XMITACLK3 (SFI15 Transmit: Test Point A Clock Signal 3, 3.125 Gb/s)  
 SFI5\_XMITCCLK3 (SFI15 Transmit: Test Point C Clock Signal 3, 3.125 Gb/s)  
 SFI5\_RCVBDATA3 (SFI15 Receive: Test Point B Data Signal 3, 3.125 Gb/s)  
 SFI5\_RCVDDATA3 (SFI15 Receive: Test Point D Data Signal 3, 3.125 Gb/s)

SFI5\_RCVBCLK3 (SFI15 Receive: Test Point B Clock Signal 3, 3.125 Gb/s)  
 SFI5\_RCVDCLK3 (SFI15 Receive: Test Point D Clock Signal 3, 3.125 Gb/s)  
 STM1E\_0 (ITU-T, G703 (10/98), STM1E Binary 0)  
 STM1E\_1 (ITU-T, G703 (10/98), STM1E Binary 1)  
 STS1Pulse (ANSI T1.102-1993 (R1999), STS-1 Pulse, 51.84 Mb/s)  
 STS1Eye (ANSI T1.102-1993 (R1999), STS-1 Eye, 51.84 Mb/s)  
 STS3 (ANSI T1.102-1993 (R1999), STS-3, 155.52 Mb/s)  
 STS3\_Max (ANSI T1.102-1993 (R1999), STS-3 Max Output, 155.52 Mb/s)  
 TFI15\_2 (TFI-5, 2.488 Gb/s)  
 TFI15\_3 (TFI-5, 3.1104 Gb/s)  
 USERMask  
 VIDEO292M (SMPTE, 1.485 Gb/s)  
 VSROC192 (VSR OC192/STM64, 1.24416 Gb/s)

**Examples** MASK:AUTOSET:STANDARDE1COAX selects the E1 Coax standard mask.  
 MASK:AUTOSET:STANDARD? might return :MASK:AUTOSET:STANDARD  
 E1COAX, indicating that the E1 Coax mask is selected as the current mask standard.

## MASK:AUTOSet:TRIGger

This command specifies whether a mask [AUTOSet](#) affects the trigger level. Other trigger settings such as type of trigger are not changed by autose. The default is ON.

This command, like all the mask autose commands, affects only an autose on a standard mask, not the general instrument autose function.

**Group** Mask

**Syntax** MASK:AUTOSet:TRIGger {ON|OFF|<NR1>}  
 MASK:AUTOSet:TRIGger?

**Arguments** ON enables the autose:trigger function.  
 OFF disables the autose:trigger function.  
 <NR1> = 0 disables the autose:trigger function; any other value enables it.

**Examples** MASK:AUTOSET:TRIGGERON enables the autose:trigger function.  
 MASK:AUTOSET:TRIGGER? might return  
 :MASK:AUTOSET:TRIGGER 1, indicating that the autose:trigger function is enabled.

## MASK:AUTOSet:USER:ONE

This command sets or queries the level 1 value used by autose for user masks.

**Group** Mask

**Syntax** MASK:AUTOSet:USER:ONE <NR3>  
 MASK:AUTOSet:USER:ONE?

**Related Commands** [MASK:AUTOSet:USER:TYPe](#), [MASK:AUTOSet:USER:ZERo](#)

**Arguments** <NR3> sets the level 1 value used by autose for user masks.

**Examples** MASK:AUTOSET:USER:ONE1 sets the autose for 1 in user masks is 1.0 volts.  
 MASK:AUTOSET:USER:ONE? might return :MASK:AUTOSET:USER:ONE 0.0000, indicating that the autose is using 0.0 volts for 1 in user masks.

## MASK:AUTOSet:USER:TYPe

This command sets or queries how autose scales a mask.

**Group** Mask

**Syntax** MASK:AUTOSet:USER:TYPe {ABSolute|NORMALIZED}  
 MASK:AUTOSet:USER:TYPe?

**Related Commands** [MASK:AUTOSet:USER:ONE](#), [MASK:AUTOSet:USER:ZERo](#)

**Arguments** ABSOLUTE tells autose to use the values supplied by [MASK:AUTOSet:USER:ONE](#) and [MASK:AUTOSet:USER:ZERo](#) for the one and zero of the mask.

NORMALIZED tells autoseT to determine the mask one and zero by looking at the mask.

**Examples** MASK:AUTOSET:USER:TYPE NORMALIZED sets the autoseT scales in a mask to normalize.

MASK:AUTOSET:USER:TYPE? might return :MASK:AUTOSET:USER:TYPE NORMALIZED, indicating that the autoseT is set to normalize scales in a mask.

## MASK:AUTOSet:USER:ZERo

This command sets or queries the level 0 used by autoseT for user masks.

**Group** Mask

**Syntax** MASK:AUTOSet:USER:ZERo <NR3>  
MASK:AUTOSet:USER:ZERo?

**Related Commands** [MASK:AUTOSet:USER:TYPE](#), [MASK:AUTOSet:USER:ONE](#)

**Arguments** <NR3> sets the zero level that autoseT uses for user masks.

**Examples** MASK:AUTOSET:USER:ZERo1.000 sets the autoseT for zero in user masks to scales in a mask to normalize.

MASK:AUTOSET:USER:ZERo? might return :MASK:AUTOSET:USER:ZERo 0.0000, indicating that the autoseT is using 0.0 volts for zero in user masks.

## MASK:AUTOSet:VPOS

This command controls whether the mask autoseT algorithm will affect the vertical position (or offset) of the signal while attempting to autoseT. The default is ON.

This command, like all the mask autoseT commands, affects only an autoseT on a standard mask, not the general instrument autoseT function.

**Group** Mask

**Syntax** MASK:AUTOSet:VPOS {ON|OFF|<NR1>}  
MASK:AUTOSet:VPOS?



**Arguments** ON enables the autosest:vpos function.  
 OFF disables the autosest:vpos function.  
 <NR1> = 0 disables the autosest:vpos function; any other value enables it.

**Examples** MASK:AUTOSET:VPOS ON enables the autosest:vpos function.  
 MASK:AUTOSET:VPOS? might return :MASK:AUTOSET:VPOS 0.0000,  
 indicating that the autosest:vpos function is disabled.

## MASK:AUTOSet:VSCAlE

This command controls whether the mask autosest algorithm will affect the vertical scale while attempting to autosest.

This command, like all the mask autosest commands, affects only an autosest on a standard mask, not the general instrument autosest function.

**Group** Mask

**Syntax** MASK:AUTOSet:VSCAlE {ON|OFF|<NR1>}  
 MASK:AUTOSet:VSCAlE?

**Arguments** ON enables the autosest:vscale function.  
 OFF disables the autosest:vscale function.  
 <NR1> = 0 disables the autosest:vscale function; any other value enables it.

**Examples** MASK:AUTOSET:VSCALE ON enables the autosest:vscale function.  
 MASK:AUTOSET:VSCALE? might return :MASK:AUTOSET:VSCALE 0.0000,  
 indicating that the autosest:vscale function is disabled.

## MASK:COpy:USER (No Query Form)

This command copies the currently displayed mask to the user mask.

**Group** Mask

**Syntax** MASK:COpy:USER

**Examples** MASK:COPY:USER copies the current mask to the user mask.

## MASK:COUNT (No Query Form)

This command resets to zero the number of hits for all mask segments on the specified source.

This is equivalent to pressing the front panel 'Clear' button.

**Group** Mask

**Syntax** MASK:COUNT RESET

**Arguments** RESET sets to zero the number of hits for all mask segments on the specified source.

**Examples** MASK:COUNT resets to zero the number of hits for all mask segments on the specified source.

## MASK:COUNT:FAILURES? (Query Only)

This query-only command returns the number of pass/fail tests that have failed. The default is 0. Pass/fail testing must be enabled.

**Group** Mask

**Syntax** MASK:COUNT:FAILURES?

**Returns** <NR1>

**Examples** MASK:COUNT:FAILURES? might return :MASK:COUNT:FAILURES 0, showing that no tests have failed.

## MASK:COUNT:HITS? (Query Only)

This query-only command returns the number of hits on all mask segments. Hit counting must be turned on.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:COUNT:HITS?</code>
<b>Returns</b>	<code>&lt;NR1&gt;</code>
<b>Examples</b>	<code>MASK:COUNT:HITS?</code> might return <code>:MASK:COUNT:HITS 0</code> , showing there were no hits from any source on any mask segments.

### **MASK:COUNT:SEG<m>:HITS? (Query Only)**

This query-only command returns the number of hits in mask segment `<m>`. Hit counting must be turned on.

`m` is the mask segment number.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:COUNT:SEG&lt;m&gt;:HITS?</code>
<b>Returns</b>	<code>&lt;NR1&gt;</code> returns the number of hits in the mask segment.
<b>Examples</b>	<code>MASK:COUNT:SEG1:HITS?</code> might return <code>:MASK:COUNT:SEG1:HITS 0</code> , showing that there have been no hits.

### **MASK:COUNT:STATE**

This command sets or queries the mask hits count state; it controls whether mask counting is being done. `MASK:DISplay` must be ON to enable `MASK:COUNT:STATE` to count mask violations.

<b>Group</b>	Mask
<b>Syntax</b>	<code>MASK:COUNT:STATE {OFF ON &lt;NR1&gt;}</code> <code>MASK:COUNT:STATE?</code>
<b>Arguments</b>	ON turns on mask counting. OFF turns off mask counting. This is the default state.

<NR1> = 0 turns off mask hit counting, and other values turn on mask hit counting.

**Examples** MASK:COUNT:STATEON turns on mask hit counting.

MASK:COUNT:STATE? might return :MASK:COUNT:STATE 1, indicating that the mask counting is on.

## MASK:COUNT:TESTS? (Query Only)

This query-only command returns the number of pass/fail tests that have occurred.

**Group** Mask

**Syntax** MASK:COUNT:TESTS?

**Returns** <NR1>

**Examples** MASK:COUNT:TESTS? might return :MASK:COUNT:TESTS 50, showing 50 tests have occurred.

## MASK:COUNT:TOTAL? (Query Only)

This query-only command returns the sum of all hits in all mask segments. This command is the same as MASK:COUNT:HITS? and is kept for compatibility with other Tektronix instruments.

**Group** Mask

**Syntax** MASK:COUNT:TOTAL?

**Returns** <NR1>

**Examples** MASK:COUNT:TOTAL? might return :MASK:COUNT:TOTAL50, showing 50 hits have occurred.

## MASK:COUNT:VIOLATIONS? (Query Only)

This query-only command returns the number of test violations that have occurred in the current pass/fail test. A test violation occurs when any part of a waveform falls within any mask segment. The default is 0.

**Group** Mask

**Syntax** MASK:COUNT:VIOLATIONS?

**Related Commands** [MASK:COUNT:TESTS?](#), [MASK:COUNT:TOTAL?](#), [MASK:COUNT:WAVEFORMS?](#), [MASK:TEST:REPEAT](#)

**Returns** <NR1>

**Examples** MASK:COUNT:VIOLATIONS? might return :MASK:COUNT:VIOLATIONS 50, showing that 50 test violations have occurred.

## MASK:COUNT:WAVEFORMS? (Query Only)

This query-only command returns the number of waveforms that have been acquired and processed during pass/fail testing.

**Group** Mask

**Syntax** MASK:COUNT:WAVEFORMS?

**Related Commands** [MASK:COUNT:TESTS?](#), [MASK:COUNT:TOTAL?](#), [MASK:COUNT:VIOLATIONS?](#)

**Returns** <NR1>

**Examples** MASK:COUNT:WAVEFORMS? might return :MASK:COUNT:WAVEFORMS 1000, showing that 1,000 waveforms have been acquired and processed.

## MASK:DISplay

This command sets or queries whether or not defined masks are displayed on the screen. This is useful for temporarily turning off user-defined masks without deleting them. It is also useful for removing a standard mask from the screen, but leaving it as the selected standard. Mask counting, mask testing, and mask autoset are unavailable if the mask display is Off. The default is On.

**Group** Mask

**Syntax** MASK:DISplay {OFF|ON|<NR1>}  
MASK:DISplay?

**Arguments** ON shows the masks on the display. This is the default value.  
OFF removes the masks from the display.  
<NR1> = 0 removes the masks from the display; any other value shows the masks on the display.

**Examples** MASK:DISPLAYON sets the display to show the defined masks.  
MASK:DISPLAY? might return :MASK:DISPLAY 1, indicating that the defined masks on the screen are displayed.

## MASK:FILTer

This command sets or returns whether the selected digital filter will be run on the waveform data. The filter simulates optical hardware. That is, it simulates different hardware for each of several different optical standards.

The digital filter runs on OC1, OC3, OC12, OC48, FC133, FC266, FC531, FC1063, FC2125Draft, Gigabit Ethernet, Infiniband 2.5Gb, 1394b, 393Mb, 786.4 3Mb, 1.572 Gb

**Group** Mask

**Syntax** MASK:FILTer {OFF|ON|<NR1>}  
MASK:FILTer?

**Related Commands** [ACQuire:MODE](#)

**Arguments** OFF disables the digital filter.  
 ON enables the digital filter.  
 <NR1> = 0 disables the digital filter; any other value enables it.

**Examples** MASK:FILTERON enables the digital filter.  
 MASK:FILTER? might return :MASK:FILTER 1, indicating that the selected digital filter will be run on the waveform data.

## MASK:HIGHLIGHTHits

This command sets or returns whether or not hits in a mask are highlighted in different colors than other waveform data. The default is On.

**Group** Mask

**Syntax** MASK:HIGHLIGHTHits {OFF|ON|<NR1>}  
 MASK:HIGHLIGHTHits?

**Arguments** OFF disables the mask:highlighthits function.  
 ON enables the mask:highlighthits function.  
 <NR1> = 0 disables the mask:highlighthits function; any other value enables it.

**Examples** MASK:HIGHLIGHTHITSON enables the mask highlighthits function.  
 MASK:HIGHLIGHTHITS? might return :MASK:HIGHLIGHTHITS 1, indicating that the mask:highlighthits function is enabled.

## MASK:INVert

This command controls whether the mask is drawn inverted. It has no effect if this mask cannot be inverted. The default is Off (Positive).

**Group** Mask

**Syntax** MASK:INVert {OFF|ON|<NR1>}

<b>Arguments</b>	<p>OFF Positive</p> <p>ON Negative</p> <p>&lt;NR1&gt; = 0 disables the mask:invert function; any other value enables it.</p>
<b>Examples</b>	<p>MASK:INVERTON enables the mask invert function.</p> <p>MASK:INVERT? might return :MASK:INVERT 1, indicating that the mask is drawn inverted.</p>

## MASK:LOCK

This command sets or returns the mask lock to waveform mode. It locks the mask to the waveform such that changes to the horizontal and/or vertical scale settings redraws the mask segments in proportion. This feature allows for expanding horizontal and/or vertical settings to zoom in on waveforms and masks to visually examine violation areas in more detail.

<b>Group</b>	Mask
<b>Syntax</b>	<p>MASK:LOCK {OFF ON &lt;NR1&gt;}</p> <p>MASK:LOCK?</p>
<b>Arguments</b>	<p>OFF turns off mask lock to waveform.</p> <p>ON turns on mask lock to waveform.</p> <p>&lt;NR1&gt; = 0 turns off mask lock to waveform and any other value turns on mask lock to waveform.</p>
<b>Examples</b>	<p>MASK:LOCKON turns on mask lock to waveform.</p> <p>MASK:LOCK? might return :MASK:LOCK 1, indicating that the mask lock to waveform mode is enabled.</p>

## MASK:MARgin:PERCent

This command sets or queries the mask margin percentage. If you set a positive mask margin percentage, the polygon expands and the mask test will be more difficult to pass due to the expanded test area. If you set a negative mask margin percentage, the polygon shrinks and the mask test will be easier to pass due to the smaller area.



<b>Group</b>	Mask
<b>Syntax</b>	MASK:MARgin:PERCent <NR3> MASK:MARgin:PERCent?
<b>Arguments</b>	<NR3> values range from -50.0 to +50.0. The default is 5.
<b>Examples</b>	MASK:MARGIN:PERCENT25 sets the mask margin percentage to 25.  MASK:MARGIN:PERCENT? might return :MASK:MARGIN:PERCENT 1.0000000000E+001, indicating that the mask margin is set to 10%.

## MASK:MARgin:STATE

This command sets or returns the state of the mask margins.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:MARgin:STATE {OFF ON <NR1>} MASK:MARgin:STATE?
<b>Arguments</b>	OFF turns off mask margins. The currently displayed margined mask is erased and the original mask is displayed.  ON turns on mask margins.  <NR1> = 0 turns off mask margins; any other integer turns on the selected mask margins.
<b>Examples</b>	MASK:MARGIN:STATEON turns on mask margins.  MASK:MARGIN:STATE? might return :MASK:MARGIN:STATE 1, indicating that the mask margins are on.

## MASK:MASKPRE:AMPLitude

This command sets or returns the current mask's nominal pulse amplitude in volts. The query form of this command returns the nominal pulse amplitude of the displayed mask. The set form of this command only affects the current (displayed) mask.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:MASKPRE:AMPLitude <NR3> MASK:MASKPRE:AMPLitude?
<b>Related Commands</b>	<a href="#">MASK:MASKPRE:VSCale</a>
<b>Arguments</b>	<NR3> is a floating number that sets the nominal pulse amplitude in volts.
<b>Examples</b>	MASK:MASKPRE:AMPLITUDE2.4 sets the current mask nominal pulse amplitude to 2.4 volts.  MASK:MASKPRE:AMPLITUDE? might return :MASK:MASKPRE:AMPLITUDE 1, indicating that the current mask's nominal pulse amplitude is set to 1.0 V.

## MASK:MASKPRE:HSCALE

This command sets or returns the nominal timing resolution used to draw the mask, in time/division. The query form of this command returns the nominal timing resolution of the displayed mask. The set form of this command only affects the current (displayed) mask.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:MASKPRE:HSCALE <NR3> MASK:MASKPRE:HSCALE?
<b>Related Commands</b>	<a href="#">MASK:MASKPRE:HTRIGPOS</a>
<b>Arguments</b>	<NR3> is a floating point number that sets the mask drawing timing resolution.
<b>Examples</b>	MASK:MASKPRE:HSCALE40E-9 sets the nominal timing resolution used to draw the mask pulse shape to 40 nanoseconds per division.  MASK:MASKPRE:HSCALE? might return :MASK:MASKPRE:HSCALE 1, indicating that the current mask's nominal timing resolution is to 1 nanosecond per divisions.

## MASK:MASKPRE:HTRIGPOS

This command sets or returns the nominal trigger position (pulse leading edge) used to draw the mask as a fraction of the display width. The query form of this command returns the nominal trigger position of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:HTRIGPOS <NR3>  
MASK:MASKPRE:HTRIGPOS?

**Related Commands** [MASK:MASKPRE:HSCALE](#)

**Arguments** <NR3> is a floating point number in the range of 0.0 to 1.0 that sets the trigger points as a fraction of the display width, referenced from the left edge of the graticule. The number 0.0 represents the left edge.

**Examples** MASK:MASKPRE:HTRIGPOS0.6 sets the current mask nominal pulse leading edge position to the sixth horizontal graticule division.

MASK:MASKPRE:HTRIGPOS? might return :MASK:MASKPRE:HTRIGPOS 0.8, indicating that the trigger position is set to eight horizontal graticule divisions.

## MASK:MASKPRE:PATTERNBITS

This command sets or returns the number of bits used for serial trigger for the User mask standard. For example, DS1 requires six bits: four leading zeros, a one, and a trailing zero. The query form of this command returns the serial bit value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:PATTERNBITS <NR1>  
MASK:MASKPRE:PATTERNBITS?

**Related Commands** [MASK:MASKPRE:PRESAMPBITS](#)

**Arguments** <NR1> is an integer that sets the number of bits.

- Examples** MASK:MASKPRE:PATTERNBITS6 sets the current mask number of serial pattern bits to 6.
- MASK:MASKPRE:PATTERNBITS? might return :MASK:MASKPRE:PATTERNBITS 8, indicating that the number of bits for serial trigger of the user mask standard is set to 8.

## MASK:MASKPRE:PRESAMPBITS

This command sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing. For example, DS1 has four leading zeros. The query form of this command returns the presample bit value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:PRESAMPBITS <NR1>  
MASK:MASKPRE:PRESAMPBITS?

**Related Commands** [MASK:MASKPRE:PATTERNBITS](#)

**Arguments** <NR1> is an integer that sets the number of bits before the trigger pulse.

- Examples** MASK:MASKPRE:PRESAMPBITS4 sets the current mask number of bits before the trigger pulse to 4.
- MASK:MASKPRE:PRESAMPBITS? might return :MASK:MASKPRE:PRESAMPBITS 8, indicating that the presample bit value of the displayed mask is set to 8.

## MASK:MASKPRE:RECOrdlength

This command sets or returns the nominal record length for pulse mask testing. The query form of this command returns the record length value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:RECOrdlength <NR1>  
MASK:MASKPRE:RECOrdlength?

<b>Arguments</b>	<NR1> is an integer number that sets the record length.
<b>Examples</b>	MASK:MASKPRE:RECORDLENGTH500 sets the current mask record length to 500 points.  MASK:MASKPRE:RECORDLENGTH? might return :MASK:MASKPRE:RECORDLENGTH 400, indicating that the nominal record length for pulse mask testing is set to 400.

## MASK:MASKPRE:TRIGTOSAMP

This command sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position. The query form of this command returns the time value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:TRIGTOSAMP <NR3>  
MASK:MASKPRE:TRIGTOSAMP?

**Related Commands** [MASK:MASKPRE:WIDTH](#)

**Arguments** <NR3> is the floating point number that sets the time to the pulse bit sampling position.

**Examples** MASK:MASKPRE:TRIGTOSAMP162E-9 sets the current mask time from the (leading edge) trigger position to the pulse bit sampling position to 162 nanoseconds.  
  
MASK:MASKPRE:TRIGTOSAMP? might return :MASK:MASKPRE:TRIGTOSAMP 100, indicating that the time value of the displayed mask is set to 100 nanoseconds.

## MASK:MASKPRE:VOFFSet

This command sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels. The query form of this command returns the offset value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:VOFFSet <NR3>  
 MASK:MASKPRE:VOFFSet?

**Related Commands** [MASK:MASKPRE:VPOS](#), [MASK:MASKPRE:VSCALE](#)

**Arguments** <NR3> is a floating point number that sets the nominal vertical offset in volts.

**Examples** MASK:MASKPRE:VOFFSET0.0 sets the current mask nominal vertical offset to 0.0 volts.

MASK:MASKPRE:VOFFSET? might return :MASK:MASKPREVOFFSET 1.0000, indicating that the offset value of the displayed mask is set to 1.0 volts.

## MASK:MASKPRE:VPOS

This command sets or returns the nominal vertical position, control in divisions, used to vertically position the input channels. The query form of this command returns the vertical position value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:VPOS <NR3>  
 MASK:MASKPRE:VPOS?

**Related Commands** [MASK:MASKPRE:VOFFSet](#), [MASK:MASKPRE:VSCALE](#)

**Arguments** <NR3> is a floating point number that sets the nominal vertical position control in divisions.

**Examples** MASK:MASKPRE:VPOS-0.8 sets the current mask nominal vertical position to -0.8 divisions.

MASK:MASKPRE:VPOS? might return :MASK:MASKPRE:VPOS 1.0000, indicating that the vertical position value of the displayed mask is set to 1.0 divisions.

## MASK:MASKPRE:VSCALE

This command sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels. The query form of this command returns the

vertical scale value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:VSCALE <NR3>  
MASK:MASKPRE:VSCALE?

**Related Commands** [MASK:MASKPRE:VOFFSet](#), [MASK:MASKPRE:VPOS](#)

**Arguments** <NR3> is a floating point number that sets the nominal bit width in seconds.

**Examples** MASK:MASKPRE:VSCALE200E-3 sets the current mask nominal vertical scale to 200 millivolts per division.

MASK:MASKPRE:VSCALE? might return :MASK:MASKPRE:VSCALE 400.0000E-3, indicating that the vertical position value of the displayed mask is set to 400 millivolts per division.

## MASK:MASKPRE:WIDTH

This command sets or returns the nominal bit width in seconds. The query form of this command returns the bit width value of the displayed mask. The set form of this command only affects the current (displayed) mask.

**Group** Mask

**Syntax** MASK:MASKPRE:WIDTH <NR3>  
MASK:MASKPRE:WIDTH?

**Related Commands** [MASK:MASKPRE:TRIGTOSAMP](#)

**Arguments** <NR3> is a floating point number that sets the nominal bit width in seconds. This number is the time of one bit of data where  $\text{bit width} = 1 / (\text{data rate of the signal})$ .

**Examples** MASK:MASKPRE:WIDTH648E-9 sets the current mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

MASK:MASKPRE:WIDTH? might return :MASK:MASKPRE:WIDTH 500.0000E-9, indicating that the bit width value of the displayed mask is set to 500 nanoseconds.

## MASK:POLarity

This command sets or returns the input waveform polarity for the pass/fail test. It controls whether to test positive pulse, negative pulse, or both during pass/fail testing. This command only applies when [MASK:TEST:STATE](#) is on.

**Group** Mask

**Syntax** MASK:POLarity {BOTH|NEGATIVE|POSITIVE}  
MASK:POLarity?

**Related Commands** [MASK:TEST:STATE](#)

**Arguments** BOTH enables testing for both positive and negative pulses. The instrument tests positive pulses on the mask:source waveform until ½ of the waveform has been tested. Then the instrument inverts the mask and performs the remaining tests.

NEGATIVE enables testing on negative pulses.

POSITIVE enables testing on positive pulses. This is the default.

**Examples** MASK:POLARITYBOTH sets pass/fail to test both positive (normal) and negative (inverted) waveforms.

MASK:POLARITY? might return :MASK:POLARITY POSITIVE, indicating that the input waveform polarity for the pass/fail test is set to positive.

## MASK:SEG<m> (No Query Form)

This command deletes the specified mask segment from the current mask.

m is an integer that specifies the mask segment number to delete from the current mask.

**Group** Mask

**Syntax** MASK:SEG<m> DELETE

**Arguments** DELETE removes the specified mask segment from the mask.

**Examples** MASK:SEG3 DELETE deletes mask segment three from the current mask.



## MASK:SEG<m>:NR\_Pt? (Query Only)

This query-only command returns the number of points that make up the specified mask segment of the current mask. Each mask point consists of an X-Y pair of coordinates.

m is an integer number that specifies a mask segment number of the current mask.

**Group** Mask

**Syntax** MASK:SEG<m>:NR\_Pt?

**Returns** <NR1> is 0 to 50.

**Examples** MASK:SEG2:NR\_PT? might return :MASK:SEG2:NR\_PT 5.

## MASK:SEG<m>:POINTS

This command sets or returns the X-Y user coordinates of all points in the current mask segment. The set form defines new points in the current mask, replacing any existing points in the current mask segment; it sets or returns the vertices for a particular segment in the current mask.

m is an integer that specifies the current mask segment number.

**Group** Mask

**Syntax** MASK:SEG<m>:POINTS <NR3>, <NR3> [, <NR3>, <NR3>]  
MASK:SEG<m>:POINTS?

**Arguments** <NR3> refers to the coordinates of one of the vertices in the Current mask. Each pair of numbers represents the horizontal and vertical coordinates of a mask segment vertex. The pairs must be listed in a counterclockwise order. If the vertical or horizontal scale or position is changed after this command and then the query form of this command is issued, the value returned from the instrument will not be the same. If just one pair is input, it is ignored and the current mask segment is marked as undefined. The default is no points in the current mask segment.

**Examples** MASK:SEG2:POINTS -2.3E-9, 44E-3, -2.5E-9, 47E-3, 1.2E-9, 44E-3 defines the points in current mask segment 2.

`MASK:SEG2:POINTS?` might return `:MASK:SEG2:POINTS 0.0000, 0.0000`, indicating that the X-Y user coordinates of all points in the Current mask segment are set to 0,0.

## MASK:SOURce

This command sets or reports which source will be compared against the mask(s) when counting is turned on; it controls which trace to use in mask counting. It also affects mask autoselect and how triggering is set up when you select the mask.

**Group** Mask

**Syntax** `MASK:SOURce {CH[1-4]|MATH[1-4]|REF[1-4]}`  
`MASK:SOURce?`

**Arguments** `CH<1-4>` selects a channel waveform to be compared against the specified mask. The range for is 1 through 4.

`MATH<1-4>` selects a math waveform to be compared against the specified mask. The range for is 1 through 4.

`REF<1-4>` selects a reference waveform to be compared against the specified mask. The range is 1 through 4.

**Examples** `MASK:SOURCECH1` enables CH1 to be compared against the mask. Since a timebase is not specified, the default, Main, is used.

`MASK:SOURCE?` might return `:MASK:SOURCE CH1,MAIN`, indicating that the Channel 1 waveform will be compared against the mask using the Main timebase.

## MASK:STANdard

This command deletes the existing mask (if any) and sets the selected standard mask. If `MASK:COUNt:STATE` is ON, mask counting starts immediately. The query form of this command returns the current mask standard.

---

**NOTE.** *There are several available mask standards that require a higher bandwidth than permitted for some instruments. In such cases, a warning is posted to the standard event system upon selection of any of these standards. To ensure that the mask you select is fully supported by the instrument, always check the event status register after selecting a mask standard.*

---

The following warning event is posted if the mask exceeds the instrument bandwidth: 2318, “Consider system bandwidth when testing at this bit rate.”

**Conditions** This command is only valid for instruments with Option MTM installed.

**Group** Mask

**Syntax** MASK:STANDARD {ATARXG1|ATARXG2|ATARXG3|ATATXG1|ATATXG2|ATATXG3|CLOCKCoax|CLOCKSymmetrical|D1|D2|DS0Contra|DS0Double|DS0Single|DS0Timing|DS1|DS1A|DS1C|DS2|DS2RATECoax|DS2RATESymmetrical|DS3|DS4NA|DS4NA\_Max|E1Coax|E1Symmetrical|E2|E3|E4\_1|E4\_0|ENET1000BCX\_ATP2|ENET1000BCX\_ATP3|ENET1000BCX\_NTP2|ENET100FX|ENET100STP|ENET100UTP|ENET1250|ENETXAUI\_Far|ENETXAUI\_Near|FC1063|FC1063Draft|FC1063E|FC1063E\_ABR|FC1063E\_ABT|FC1063E\_ADR|FC1063E\_ADT|FC1063E\_AGR|FC1063E\_AGT|FC1063E\_NBT|FC1063E\_NDT|FC1063E\_NGT|FC133|FC133E|FC2125|FC2125E\_ABR|FC2125E\_ABT|FC2125E\_ADR|FC2125E\_ADT|FC2125E\_AGR|FC2125E\_AGT|FC2125E\_NBT|FC2125E\_NDT|FC2125E\_NGT|FC266|FC266E|FC4250E\_ABR|FC4250E\_ABT|FC4250E\_ADR|FC4250E\_ADT|FC4250E\_AGR|FC4250E\_AGT|FC4250E\_NBT|FC4250E\_NDT|FC4250E\_NGT|FC531|FC531E|FST1|FST2|FST3|FST4|FST5|FST6|FW1394BS1600B|FW1394BS1600BT1|FW1394BS1600BT2|FW1394BS400B|FW1394BS400BT1|FW1394BS400BT2|FW1394BS800B|FW1394BS800BT1|FW1394BS800BT2|G703DS1|G703DS3|HST1|HST2|HST3|HST4|HST5|HST6|INF2\_5G|INF2\_5GE|NONE|OC1|OC12|OC3|OC48|OC48\_FEC|PCIEXPRESS\_Rcv|PCIEXPRESS\_Xmit|RATE32Mbit|RATE97Mbit|RIO\_DRV1G|RIO\_DRV1\_5G|RIO\_DRV2G|RIO\_DRV500M|RIO\_DRV500M|RIO\_DRV750M|RIO\_EDRV1G|RIO\_EDRV1\_5G|RIO\_EDRV2G|RIO\_EDRV500M|RIO\_EDRV500M|RIO\_EDRV750M|RIO\_RCV1G|RIO\_RCV1\_5G|RIO\_RCV2G|RIO\_RCV500M|RIO\_RCV500M|RIO\_RCV750M|RIO\_SERIAL\_1G|RIO\_SERIAL\_2G|RIO\_SERIAL\_3G|SFI5\_RCVBCLK2|SFI5\_RCVBCLK3|SFI5\_RCVBDATA2|SFI5\_RCVBDATA3|SFI5\_RCVCLK2|SFI5\_RCVCLK3|SFI5\_RCVDDATA2|SFI5\_RCVDDATA3|SFI5\_XMITACLK2|SFI5\_XMITACLK3|SFI5\_XMITADATA2|SFI5\_XMITADATA3|SFI5\_XMITCCLK2|SFI5\_XMITCCLK3|SFI5\_XMITCDATA2|SFI5\_XMITCDATA3|STM0\_0|STM0\_1|STM0\_HDBX|STM1E\_1|STM1E\_0|STS1Eye|STS1Pulse|STS3|STS3\_Max|TFI5\_2|TFI5\_3|USERMask|VIDEO270|VIDEO292M|VIDEO360|VSROC192|SAS1\_5\_IR|SAS1\_5\_CR|SAS1\_5\_XR|SAS1\_5\_IR\_AASJ|SAS1\_5\_CR\_AASJ|SAS1\_5\_XR\_AASJ|SAS1\_5\_SATA|SAS3\_0\_IR|SAS3\_0\_CR|SAS3\_0\_XR|SAS3\_0\_IR\_AASJ|SAS3\_0\_CR\_AASJ|SAS3\_0\_XR\_AASJ|SAS3\_0\_SATA}

MASK:STANDARD?

**Related Commands**    [ACQUIRE:MODE](#)

**Arguments**

ATARXG1 (Serial ATA, G1 Rx 1.5 Gb/s)  
 ATARXG2 (Serial ATA, G2, Rx, 3.0 Gb/s)  
 ATARXG3 (Serial ATA, G3, Rx)  
 ATATXG1 (Serial ATA, G1 Tx, 1.5 Gb/s)  
 ATATXG2 (Serial ATA, G2 Tx, 3.0 Gb/s)  
 H ATATXG3 (Serial ATA, G3 Tx)  
 CLOCKCoax  
 CLOCKSymmetrical  
 D1  
 D2  
 DS0Contra (ITU-T, G703 (10/98), 64 kb/s)  
 DS0Double (ITU-T, G703 (10/98), 64 kb/s)  
 DS0Single (ITU-T, G703 (10/98), 64 kb/s)  
 DS0Timing (ITU-T, G703 (10/98), 64 kb/s)  
 DS1 (ANSI T1.102-1993 (R1999), DS1, 1.544 Mb/s)  
 DS1A (ANSI T1.102-1993 (R1999), DS1A, 2.048 Mb/s)  
 DS1C (ANSI T1.102-1993 (R1999), DS1C, 3.152 Mb/s)  
 DS2 (ANSI T1.102-1993 (R1999), DS2, 6.312 Mb/s)  
 DS2RATECoax (ITU-T, G703 (10/98), D2 Rate Coax, 6.312 Mb/s)  
 DS2RATESymmetrical (ITU-T, G703 (10/98), D2 Rate Sym, 6.312 Mb/s)  
 DS3 (ANSI T1.102-1993 (R1999), DS3, 44.736 Mb/s)  
 DS4NA (ANSI T1.102-1993 (R1999), DS4NA, 139.26 Mb/s)  
 DS4NA\_Max (ANSI T1.102-1993 (R1999), DSNA Max Output, 139.26 Mb)  
 E1Coax (ITU-T, G703 (10/98), E1 Coax Pair, 2.048 Mb/s)  
 E1Symmetrical (ITU-T, G703 (10/98), E1 Sym Pair, 2.048 Mb/s)  
 E2 (ITU-T, G703 (10/98), E2, 8.448 Mb/s)  
 E3 (ITU-T, G703 (10/98), E3, 34.368 Mb/s)

E4\_0 (ITU-T, G703 (10/98), E4 Binary 0)

E4\_1 (ITU-T, G703 (10/98), E4 Binary 1)

ENET100FX

ENET100STP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, STP, 125 Mb/s )

ENET100UTP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, UTP, 125 Mb/s)

ENET1000BCX\_ATP2 (1000B-CX Abs, TP2, 1.25 Gb/s)

ENET1000BCX\_ATP3 (1000B-CX Abs, TP3, 1.25 Gb/s)

ENET1000BCX\_NTP2 (1000B-CX Norm, TP2, 1.25 Gb/s)

ENET1250 (IEEE Std 802.3 and ANSI X3.263-1995, GB Ethernet, 1.25 Gb/s)

ENETXAUI\_FAR (10 Gigabit Attachment Unit Interface (XAUI), Far, 3.125 Gb/s)

ENETXAUI\_Near (10 Gigabit Attachment Unit Interface (XAUI), Near, 3.125 Gb/s)

FC133 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 132.8 Mb/s)

FC133E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical 132.8 Mb/s)

FC266 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 265.6 Mb/s)

FC266E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 132.8 Mb/s)

FC531 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 531.2 Mb/s)

FC531E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 531.2 Mb/s)

FC1063 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 1.065 Gb/s)

FC1063E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)

FC1063E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC1063E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC1063E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC1063E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC1063E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC1063E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC1063E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)  
FC1063Draft (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, Draft Rev 11)  
FC2125 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 2.125)  
FC2125E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)  
FC2125E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)  
FC2125E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)  
FC2125E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)  
FC2125E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)  
FC2125E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)  
FC2125E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)  
FC2125E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)  
FC2125E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)  
FC4250E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Beta, Transm)  
FC4250E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Delta, Transm)  
FC4250E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Gamma, Transm)  
FC4250E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Beta, Transm)  
FC4250E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Delta, Transm)  
FC4250E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Gamma, Transm)  
FC4250E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Beta, Recv)  
FC4250E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Delta, Recv)  
FC4250E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Gamma, Recv)  
FST1 (USB, FS:T1, 12 Mb/s)  
FST2 (USB, FS:T2, 12 Mb/s)  
FST3 (USB, FS:T3, 12 Mb/s)  
FST4 (USB, FS:T4: 12 Mb/s)  
FST5 (USB, FS:T5, 12 Mb/s)  
FST6 (USB, FS:T6, 12 Mb/s)

FW1394BS400B (IEEE 1394b, S400 Optical, 491.5 Mb/s)  
FW1394BS400BT1 (IEEE 1394b, S400b T1, 491.5 Mb/s)  
FW1394BS400BT2 (IEEE 1394b, S400b T2, 491.5 Mb/s)  
FW1394BS800B (IEEE 1394b, S800 Optical, 988.0 Mb/s)  
FW1394BS800BT1 (IEEE 1394b, S800b T1, 983.0 Mb/s)  
FW1394BS800BT2 (IEEE 1394b, S800b T2, 983.0 Mb/s)  
FW1394BS1600B (IEEE 1394b, S1600 Optical, 1.966 Gb/s)  
FW1394BS1600BT1 (IEEE 1394b, S1600b T1, 1.966 Gb/s)  
FW1394BS1600BT2 (IEEE 1394b, S1600b T2, 1.966 Gb/s)  
G703D1 (ITU-T, G703 (10/98), DS1 Rate, 1.544 Mb/s)  
G703DS3 (ITU-T, G703 (10/98))  
HST1 (USB, HS:T1, 480 Mb/s) G703DS3 (ITU-T, G703 (10/98))  
HST2 (USB, HS:T2, 480 Mb/s) G703DS3 (ITU-T, G703 (10/98))  
HST3 (USB, HS:T3, 480 Mb/s)  
HST4 (USB, HS:T4, 480 Mb/s)  
HST5 (USB, HS:T5, 480 Mb/s)  
HST6 (USB, HS:T6, 480 Mb/s)  
INF2\_5G (InfiniBand, IBTA Spec 1.0a, 2.5 Optical, 2.5 Gb/s)  
INF2\_5GE (InfiniBand, IBTA Spec 1.0a, 2.5 Electrical, 2.5 Gb/s)  
NONE  
OC1 (GR 253-CORE Issue 3 9/21/2000 OC1/STM0, 51.84 Mb/s)  
OC3 (GR 253-CORE Issue 3 9/21/2000 OC1/STM1, 155.52, Mb/s)  
OC12 (GR 253-CORE Issue 3 9/21/2000 OC1/STM4, 622.08 Mb/s)  
OC48 (GR 253-CORE Issue 3 9/21/2000 OC1/STM16, 2.4883 Gb/s)  
OC48\_FEC (Forward Error Correction - CSA8000 mask, 2.666 Gb/s)  
PCIEXPRESS\_Xmit (PCI Express Transmitter, 2.5 Gb/s)  
PCIEXPRESS\_Rcv (PCI Express Receiver, 2.5 Gb/s)  
RATE32Mbit (ITU-T, G703 (10/98), 32.064 Mb/s)  
RATE97Mbit (ITU-T, G703 (10/98), 97 Mbit, 97.728 Mb/s)  
RIO\_DRV1G (Rapid IO Driver, 1 Gb/s)  
RIO\_DRV1\_5G (Rapid IO Driver, 5 Gb/s)

RIO\_DRV2G (Rapid IO Driver, 2 Gb/s)  
RIO\_DRV500M (Rapid IO Driver, 500 Mb/s)  
RIO\_DRV750M (Rapid IO Driver, 750 Mb/s)  
RIO\_EDRV1G (Rapid IO Extended Driver, 1 Gb/s)  
RIO\_EDRV1\_5G (Rapid IO Extended Driver, 1.5 Gb/s)  
RIO\_EDRV2G (Rapid IO Extended Driver, 2 Gb/s)  
RIO\_EDRV500M (Rapid IO Extended Driver, 500 Mb/s)  
RIO\_EDRV750M (Rapid IO Extended Driver, 750 Mb/s)  
RIO\_RCV500M (Rapid IO Receiver, 500 Mb/s)  
RIO\_RCV750M (Rapid IO Receiver, 750 Mb/s)  
RIO\_RCV1G (Rapid IO Receiver, 1 Gb/s)  
RIO\_RCV1\_5G (Rapid IO Receiver, 1.5 Gb/s)  
RIO\_RCV2G (Rapid IO Receiver, 2 Gb/s)  
RIO\_SERIAL\_1G (Rapid IO Serial, 1.25 Gb/s)  
RIO\_SERIAL\_2G (Rapid IO Serial, 2.5 Gb/s)  
RIO\_SERIAL\_3G (Rapid IO Serial, 3.25 Gb/s)  
SFI5\_XMITADATA2 (SFI15 Transmit: Test Point A Data Signal 2, 2.488 Gb/s)  
SFI5\_XMITCDATA2 (SFI15 Transmit: Test Point C Data Signal 2, 2.488 Gb/s)  
SFI5\_XMITACLK2 (SFI15 Transmit: Test Point A Clock Signal 2, 2.488 Gb/s)  
SFI5\_XMITCCLK2 (SFI15 Transmit: Test Point C Clock Signal 2, 2.488 Gb/s)  
SFI5\_RCVBDATA2 (SFI15 Receive: Test Point B Data Signal 2, 2.488 Gb/s)  
SFI5\_RCVDDATA2 (SFI15 Receive: Test Point D Data Signal 2, 2.488 Gb/s)  
SFI5\_RCVBCLK2 (SFI15 Receive: Test Point B Clock Signal 2, 2.488 Gb/s)  
SFI5\_RCVDCLK2 (SFI15 Receive: Test Point D Clock Signal 2, 2.488 Gb/s)  
SFI5\_XMITADATA3 (SFI15 Transmit: Test Point A Data Signal 3, 3.125 Gb/s)  
SFI5\_XMITCDATA3 (SFI15 Transmit: Test Point C Data Signal 3, 3.125 Gb/s)  
SFI5\_XMITACLK3 (SFI15 Transmit: Test Point A Clock Signal 3, 3.125 Gb/s)  
SFI5\_XMITCCLK3 (SFI15 Transmit: Test Point C Clock Signal 3, 3.125 Gb/s)  
SFI5\_RCVBDATA3 (SFI15 Receive: Test Point B Data Signal 3, 3.125 Gb/s)  
SFI5\_RCVDDATA3 (SFI15 Receive: Test Point D Data Signal 3, 3.125 Gb/s)  
SFI5\_RCVBCLK3 (SFI15 Receive: Test Point B Clock Signal 3, 3.125 Gb/s)



SFI5\_RCVCLK3 (SFI15 Receive: Test Point D Clock Signal 3, 3.125 Gb/s)  
 STM0\_0 (ITU-T, G703 (10/98), STM1E Binary 0)  
 STM0\_1 (ITU-T, G703 (10/98), STM1E Binary 1)  
 STM0\_HDBX  
 STS1Eye (ANSI T1.102-1993 (R1999), STS-1 Eye, 51.84 Mb/s)  
 STS1Pulse (ANSI T1.102-1993 (R1999), STS-1 Pulse, 51.84 Mb/s)  
 STS3 (ANSI T1.102-1993 (R1999), STS-3, 155.52 Mb/s)  
 STS3\_Max (ANSI T1.102-1993 (R1999), STS-3 Max Output, 155.52 Mb/s)  
 TFI15\_2 (TFI-5, 2.488 Gb/s)  
 TFI15\_3 (TFI-5, 3.1104 Gb/s)  
 USERMask  
 VIDEO270  
 VIDEO292M  
 VIDEO360  
 VSROC192 (VSR OC192/STM64, 1.24416 Gb/s)

**Examples** MASK:STANDARDE1COAX selects the E1 Coax standard mask.  
 MASK:STANDARD? might return :MASK:STANDARD E1COAX, indicating that the E1 Coax mask is selected as the current mask standard.

## MASK:STOPOnviolation

This command sets or returns the stop on violation setting. When enabled, this command stops waveform acquisitions on the first occurrence of a waveform violation. [MASK:DISplay](#) must be ON for the instrument to detect violations.

**Group** Mask

**Syntax** MASK:STOPOnviolation {OFF|ON|<NR1>}  
 MASK:STOPOnviolation?

**Arguments** OFF turns off mask test stop on violation.  
 ON stops waveform acquisition on the first occurrence of a mask violation.  
 <NR1> = 0 turns off stop on violation; any other value turns on stop on violation.

- Examples** MASK:STOPONVIOLATIONON sets the instrument to stop waveform acquisition on the first occurrence of a waveform violation.
- MASK:STOPONVIOLATION? might return :MASK:STOPONVIOLATION ON, indicating that the stops waveform acquisitions on the first occurrence of a waveform violation is enabled.

## MASK:TEST:AUX:COMPLetion

This command sets or returns the test:aux on pass/fail test completion mode. When enabled, this command causes the instrument to provide a TTL signal at the Aux:Out port when the mask pass/fail status changes to 'Failing'.

- Group** Mask
- Syntax** MASK:TEST:AUX:COMPLetion {OFF|ON|<NR1>}  
MASK:TEST:AUX:COMPLetion?
- Arguments** OFF turns off the pass/fail mask:test:aux on completion.  
ON turns on the pass/fail mask:test:aux on completion.  
<NR1> = 0 disables the mask:test:aux:completion function; any other value enables it.
- Examples** MASK:TEST:AUX:COMPLETIONON turns on the pass/fail test on completion.  
MASK:TEST:AUX:COMPLETION? might return :MASK:TEST:AUX:COMPLETION ON, indicating that the test:aux on pass/fail test completion mode is turned on.

## MASK:TEST:AUX:FAILure

This command sets or returns the test:aux status on pass/fail test failure mode. When enabled, this command causes the instrument to provide a TTL signal at the AuxOut port when the pass/fail status changes to 'Failing'.

- Group** Mask
- Syntax** MASK:TEST:AUX:FAILure {OFF|ON|<NR1>}  
MASK:TEST:AUX:FAILure?

**Arguments** OFF turns off the pass/fail mask:test:aux on failure. This is the default.  
 ON turns on the pass/fail mask:test:aux on failure.  
 <NR1> = 0 disables the mask:test:aux:failure function; any other value enables it.

**Examples** MASK:TEST:AUX:FAILUREON turns on the pass/fail test on failure.  
 MASK:TEST:AUX:FAILURE? might return :MASK:TEST:AUX:FAILURE ON, indicating that the pass/fail mask:test:aux on failure is turned on.

## MASK:TEST:BEEP:COMPLetion

This command sets or returns the beep on pass/fail test completion mode. When enabled, this command causes the instrument to emit a tone when the mask pass/fail test completes.

**Group** Mask

**Syntax** MASK:TEST:BEEP:COMPLetion {OFF|ON|<NR1>}  
 MASK:TEST:BEEP:COMPLetion?

**Arguments** OFF turns off the pass/fail beep on completion.  
 ON turns on the pass/fail beep on completion.  
 <NR1> = 0 turns off the pass/fail beep on completion, any other integer turns on the pass/fail beep on completion.

**Examples** MASK:TEST:BEEP:COMPLETIONON turns on pas/fail beep on completion.  
 MASK:TEST:BEEP:COMPLETION? might return :MASK:TEST:BEEP:COMPLETION ON, indicating that the pass/fail beep on completion is turned on.

## MASK:TEST:BEEP:FAILure

This command sets or returns the beep status on pass/fail test failure mode. When enabled, this command causes the instrument to emit a tone when the pass/fail status changes to 'Failing'.

**Group** Mask

**Syntax** MASK:TEST:BEEP:FAILure {OFF|ON|<NR1>}  
 MASK:TEST:BEEP:FAILure?

**Arguments** OFF turns off the pass/fail beep on failure. This is the default.  
 ON turns on the pass/fail beep on failure.  
 <NR1> = 0 turns off the pass/fail beep on failure, and any other integer turns on the pass/fail beep on failure.

**Examples** MASK:TEST:BEEP:FAILUREOFF turns off the pass/fail beep on failure.  
 MASK:TEST:BEEP:FAILURE? might return :MASK:TEST:BEEP:FAILURE ON, indicating that the pass/fail beep on failure is turned on.

## MASK:TEST:DELay

This command sets or returns the amount of time, in seconds, to wait after starting pass/fail testing to evaluate waveforms. This is useful if the test system requires 'settling' time before evaluating waveforms.

**Group** Mask

**Syntax** MASK:TEST:DELay <NR3>  
 MASK:TEST:DELay?

**Arguments** <NR3> is a floating point number that specifies the amount of time, in seconds, to delay the start of the pass/fail test.

**Examples** MASK:TEST:DELAY2.5 delays waveform evaluation by 2.5 seconds after the start of pass/fail testing.  
 MASK:TEST:DELAY? might return :MASK:TEST:DELAY 5, indicating that the amount of time to wait after starting pass/fail testing to evaluate waveforms is set to 5 seconds.

## MASK:TEST:HARDCopy

This command sets or returns the hard copy in pass/fail failure mode. When enabled, this command causes the instrument to generate a screen hard copy, to the default printer using the current instrument hard copy settings, as soon as the pass/fail status changes to 'Failing'.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:TEST:HARDCopy {OFF ON <NR1>} MASK:TEST:HARDCopy?
<b>Arguments</b>	<p>OFF turns off generate hard copy on failure.</p> <p>ON turns on generate hard copy on failure.</p> <p>&lt;NR1&gt; = 0 turns off generate hard copy on failure, and any other value turns on generate hard copy on failure.</p>
<b>Examples</b>	<p>MASK:TEST:HARDCOPYON turns on generate hard copy on failure.</p> <p>MASK:TEST:HARDCOPY? might return :MASK:TEST:HARDCOPY ON, indicating that the generate hard copy on failure is turned on.</p>

## MASK:TEST:LOG:FAILure

This command sets or returns the log status on pass/fail test failure mode. When enabled, this command causes the instrument to log the current date and time to a file when the pass/fail status changes to 'Failing'. The file name is determined from the current date and time. The user cannot change this name.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:TEST:LOG:FAILure {OFF ON <NR1>} MASK:TEST:LOG:FAILure?
<b>Related Commands</b>	<a href="#">MASK:TEST:SAVEWFM:FILEName</a>
<b>Arguments</b>	<p>OFF turns off the pass/fail log on failure. This is the default.</p> <p>ON turns on the pass/fail log on failure.</p> <p>&lt;NR1&gt; = 0 turns off the pass/fail log on failure, and any other integer turns on the pass/fail log on failure.</p>
<b>Examples</b>	<p>MASK:TEST:LOG:FAILUREOFF turns off the pass/fail log on failure.</p> <p>MASK:TEST:LOG:FAILURE? might return :MASK:TEST:LOG:FAILURE ON, indicating that the pass/fail log on failure is turned on.</p>

## MASK:TEST:REPeat

This command sets or returns the repeat pass/fail test on completion mode. When enabled, this command repeats the pass/fail test cycle at the completion of the previous test cycle.

**Group** Mask

**Syntax** MASK:TEST:REPeat {OFF|ON|<NR1>}  
MASK:TEST:REPeat?

**Related Commands** [MASK:COUNt:VIOLATIONS?](#)

**Arguments** OFF turns off repeat on completion.  
ON turns on repeat on completion.  
<NR1> = 0 turns off repeat on completion, and any other integer turns on repeat on completion.

**Examples** MASK:TEST:REPEATON causes the instrument to continuously repeat the pass/fail test.  
MASK:TEST:REPEAT? might return :MASK:TEST:REPEAT ON, indicating that the repeat pass/fail test on completion is turned on.

## MASK:TEST:SAMple

This command sets or returns the minimum number of waveform database (WfmDB) points the instrument can acquire before it stops a single sequence acquisition or stops running a mask test.

Hint: a bigger sample size may allow a greater throughput (more waveforms to get acquired at a time). However, a very large sample size may slow down the display update. So if you want to speed up the display rate, consider reducing the sample size.

This command works the same as the [ACQuire:NUMSAMPles](#) command.

**Group** Mask

**Syntax** MASK:TEST:SAMple {<NR1>}  
MASK:TEST:SAMple?

**Arguments** <NR1> the number of points to sample.

**Examples** MASK:TEST:SAMPLE100000 sets the instrument to acquire 100,000 WfmDB points.

MASK:TEST:SAMPLE? might return :MASK:TEST:SAMPLE 100000, indicating that the repeat pass/fail test on completion is set to 100,000 WfmDB points.

## MASK:TEST:SAMple:THReshold

This command sets or returns the minimum number of hits in mask regions needed to cause the pass/fail status to change from PASSING to FAILING. This affects the mask test when the instrument is acquiring in waveform database (WfmDB) mode.

**Group** Mask

**Syntax** MASK:TEST:SAMple:THReshold {<NR1>}  
MASK:TEST:SAMple:THReshold?

**Arguments** <NR1> is the number of hits that can happen when the test fails.

**Examples** MASK:TEST:SAMPLE:THRESHOLD100 sets the test to fail when 100 or more hits happen.

MASK:TEST:SAMPLE:THRESHOLD? might return :MASK:TEST:THRESHOLD 1000, indicating that the repeat pass/fail test on completion is set to 1000 WfmDB points.

## MASK:TEST:SAVEWFM

This command sets or returns the save waveform to file on failure setting. When enabled, this command copies waveform data from active channels to files when the pass/fail test status changes to 'Failing'.

---

**NOTE.** *This instrument saves files only on the first violation that causes the pass/fail test to fail. To generate a waveform file for every violation occurrence, set **MASK:TEST:REPeat** to ON and **MASK:TEST:WAVEform** to 1. Use this technique only when the expected number of violations is small and the violation occurs intermittently.*

---

<b>Group</b>	Mask
<b>Syntax</b>	MASK:TEST:SAVEWFM {OFF ON <NR1>} MASK:TEST:SAVEWFM?
<b>Arguments</b>	OFF turns off copying all active channel waveforms to files. ON turns on copying all active channel waveforms to files. <NR1> = 0 turns off copying all active channel waveforms to files, and any other integer turns on copying all active channel waveforms to files.
<b>Examples</b>	MASK:TEST:SAVEWFMON sets the instrument to save the failing waveform data to files. MASK:TEST:SAVEWFM? might return :MASK:TEST:SAVEWFM 1000, indicating that the repeat pass/fail test on completion is set to 1000 WfmDB points.

## MASK:TEST:SAVEWFM:FILENAME

This command sets or returns the name of the directory and file to use with the [MASK:TEST:SAVEWFM](#) command. It defines the directory the files will be put in (name comes from date and time).

<b>Group</b>	Mask
<b>Syntax</b>	MASK:TEST:SAVEWFM:FILENAME <QString> MASK:TEST:SAVEWFM:FILENAME?
<b>Related Commands</b>	<a href="#">MASK:TEST:SAVEWFM</a> , <a href="#">MASK:TEST:LOG:FAILURE</a>
<b>Arguments</b>	<QString> is a string representing the name of the file to save waveform data to.
<b>Examples</b>	MASK:TEST:SAVEWFM:FILENAME "TEST" sets the file "TEST" as the file to save waveform data to on failure. MASK:TEST:SAVEWFM:FILENAME? might return :MASK:TEST:SAVEWFM:FILENAME "C:/TekScope/waveforms", indicating that the directory "C:/TekScope/Waveforms" the files will be put in.



## MASK:TEST:SRQ:COMPLetion

This command sets or returns an SRQ on pass/fail test completion mode. When enabled, this command causes the instrument to send a GPIB SRQ command when the mask pass/fail test completes.

**Group** Mask

**Syntax** MASK:TEST:SRQ:COMPLetion {OFF|ON|<NR1>}  
MASK:TEST:SRQ:COMPLetion?

**Arguments** OFF turns off the pass/fail SRQ on completion.  
ON turns on the pass/fail SRQ on completion.  
<NR1> = 0 turns off the pass/fail SR on completion, any other integer turns on the pass/fail SRQ on completion.

**Examples** MASK:TEST:SRQ:COMPLETIONON turns on pass/fail SRQ on completion.  
MASK:TEST:SRQ:COMPLETION? might return :MASK:TEST:SRQ:COMPLETION 1, indicating that the pass/fail SRQ on completion is turned on.

## MASK:TEST:SRQ:FAILure

This command sets or returns the SRQ status on pass/fail test failure mode. When enabled, this command causes the instrument to send a GPIB SRQ command when the pass/fail status changes to 'Failing'. In other words, it sends an SRQ when a failure occurs during the mask pass/fail test.

**Group** Mask

**Syntax** MASK:TEST:SRQ:FAILure {OFF|ON|<NR1>}  
MASK:TEST:SRQ:FAILure?

**Arguments** OFF turns off the pass/fail SRQ on failure. This is the default.  
ON turns on the pass/fail SRQ on failure.  
<NR1> = 0 turns off the pass/fail SRQ on failure, and any other integer turns on the pass/fail SRQ on failure.

**Examples** MASK:TEST:SRQ:FAILUREOFF turns off the pass/fail SRQ on failure.  
 MASK:TEST:SRQ:FAILURE? might return :MASK:TEST:SRQ:FAILURE 1, indicating that the pass/fail SRQ on failure is turned on.

## MASK:TEST:STATE

This command sets or returns the state of mask pass/fail testing. When enabled, this command resets the pass/fail status information and starts pass/fail mask testing. This command also generates an 'Operation Complete' message when pass/fail testing completes. Refer to [BUSY?](#) and [\\*OPC](#) commands for information on synchronizing a controller with pass/fail testing.

**Group** Mask

**Syntax** MASK:TEST:STATE {OFF|ON|<NR1>}  
 MASK:TEST:STATE?

**Arguments** OFF turns off mask pass/fail testing.  
 ON turns on mask pass/fail testing.  
 <NR1> = 0 turns off mask pass/fail testing; any other integer turns on mask pass/fail testing.

**Examples** MASK:TEST:STATEON turns on mask pass/fail testing.  
 MASK:TEST:STATE? might return :MASK:TEST:STATE 0, indicating that the mask pass/fail testing is turned off.

## MASK:TEST:STATUS? (Query Only)

This query-only command returns the pass/fail test status. This command returns one of: OFF, DELAY, PASSING, FAILING, PASSED, FAILED, and VIOLATION. In other words, it indicates the result of the pass/fail test. When the violation count exceeds the violation threshold, the status changes from Passing to Failed.

**Group** Mask

**Syntax** MASK:TEST:STATUS?

**Examples** MASK:TEST:STATUS? might return :MASK:TEST:STATUS PASSING indicates pass test status.

## MASK:TEST:STOP:FAILure

This command sets or returns the stop status on pass/fail test failure mode. When enabled, this command causes the instrument to stop acquiring data when the pass/fail status changes to 'Failing'. Repeat-on-completion mode has no effect.

**Group** Mask

**Syntax** MASK:TEST:STOP:FAILure {OFF|ON|<NR1>}  
MASK:TEST:STOP:FAILure?

**Arguments** OFF turns off the pass/fail stop on failure. This is the default.  
ON turns on the pass/fail stop on failure.  
<NR1> = 0 turns off the pass/fail stop on failure, and any other integer turns on the pass/fail SRQ on failure.

**Examples** MASK:TEST:STOP:FAILUREOFF turns off the pass/fail stop on failure.  
MASK:TEST:STOP:FAILURE? might return :MASK:TEST:STATE 0, indicating that the mask pass/fail testing is turned off.

## MASK:TEST:THReshold

This command sets or returns the number of failed tested waveforms needed in each pass/fail test to cause the pass/fail status to change from 'Passing' to 'Failing'.

**Group** Mask

**Syntax** MASK:TEST:THReshold <NR1>  
MASK:TEST:THReshold?

**Arguments** <NR1> is an integer that represents the number of tested waveform violations occurring in each pass/fail test needed to change the pass/fail status to 'Failing'. This means that if the threshold is set to the default of 1 then 1 failure would cause a test failure. If the threshold is set to 2 then 2 failures would cause a test failure.

- Examples** MASK:TEST:THRESHOLD20 sets the number of failed tested waveforms to 20 that is needed in each pass/fail test to cause the pass/fail status to change from 'Passing' to 'Failing'.
- MASK:TEST:THRESHOLD? might return :MASK:TEST:THRESHOLD 32, indicating that the 32 failures would cause a test failure.

## MASK:TEST:WAVEform

This command sets or returns the number of waveforms to test during pass/fail testing. If Averaging is on, this is the number of averaged waveforms to test during pass/fail testing. It specifies the number of waveforms to be acquired in counting the mask hits in the pass/fail test. The default is 20.

- Group** Mask
- Syntax** MASK:TEST:WAVEform <NR1>  
MASK:TEST:WAVEform?
- Arguments** <NR1> is an integer that sets the number of tested (sampled or averaged) waveforms to test for each pass/fail test.
- Examples** MASK:TEST:WAVEFORM20 sets the number of failures that would cause a test failure to 20.
- MASK:TEST:WAVEFORM? might return
- :MASK:TEST:WAVEFORM 50, indicating that the 50 failures would cause a test failure.

## MASK:USER:AMPLitude

This command sets or returns the current mask's nominal pulse amplitude in volts. The query form of this command returns the nominal pulse amplitude of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

- Group** Mask
- Syntax** MASK:USER:AMPLitude <NR3>  
MASK:USER:AMPLitude?

**Related Commands** [MASK:USER:VSCALE](#)

**Arguments** <NR3> is a floating number that sets the nominal pulse amplitude in volts.

**Examples** MASK:USER:AMPLITUDE2.4 sets the user mask nominal pulse amplitude to 2.4 volts.

MASK:USER:AMPLITUDE? might return

:MASK:USER:AMPLITUDE 100.0000E-3, indicating that the user mask nominal pulse amplitude to set to 0.1 volts.

## MASK:USER:BITRate

This command sets or returns the bit rate for the user mask.

**Group** Mask

**Syntax** MASK:USER:BITRate <NR1>  
MASK:USER:BITRate?

**Arguments** <NR1> is a number that sets the bit rate of the user mask in bits per second.

**Examples** MASK:USER:BITRATE156E6 sets the bit rate.

MASK:USER:BITRATE? might return :MASK:USER:BITRATE 1560000, indicating that the bit rate of the user mask is set to 1560000 bits per second.

## MASK:USER:HSCALE

This command sets or returns the nominal timing resolution used to draw the mask, in time/division. The query form of this command returns the nominal timing resolution of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:HSCALE <NR3>  
MASK:USER:HSCALE?

**Related Commands**    [MASK:USER:HTRIGPOS](#)

**Arguments**    <NR3> is a floating point number that sets the mask drawing timing resolution.

**Examples**    MASK:USER:HSCALE 200E-9 sets the nominal timing resolution used to draw the mask to 200 nanoseconds per division.

MASK:USER:HSCALE? might return :MASK:USER:HSCALE 40.0000E-9, indicating that the nominal timing resolution used to draw the mask pulse shape is set to 40 nanoseconds per division.

## MASK:USER:HTRIGPOS

This command sets or returns the nominal trigger position (pulse leading edge) used to draw the mask as a fraction of the display width. The query form of this command returns the nominal trigger position of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group**    Mask

**Syntax**    MASK:USER:HTRIGPOS <NR3>  
MASK:USER:HTRIGPOS?

**Related Commands**    [MASK:USER:HSCALE](#)

**Arguments**    <NR3> is a floating point number in the range of 0.0 to 1.0 that sets the trigger points as a fraction of the display width, referenced from the left edge of the graticule.  
  
0.0 represents the left edge.

**Examples**    MASK:USER:HTRIGPOS0.6 sets the nominal trigger position of the displayed mask to 0.6.

MASK:USER:HTRIGPOS? might return :MASK:USER:HTRIGPOS 500.0000E-3, indicating that the nominal trigger position of the displayed mask is set to 0.5.

## MASK:USER:LABel

This command sets or queries the label assigned to a user mask. This label is displayed in the upper left-hand corner of the graticule when a user mask is selected.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:USER:LABel <QString> MASK:USER:LABel?
<b>Arguments</b>	<QString> is an alphanumeric character string used to annotate a user mask label.
<b>Examples</b>	MASK:USER:LABEL "mikesdata" assigns "mikesdata" as the user mask label.  MASK:USER:LABEL? might return :MASK:USER:LABel "mikesdata", indicating that the label assigned to a user mask is set to "mikesdata".

## MASK:USER:PATTERNBITS

This command sets or returns the number of bits used for serial trigger for the User mask standard. For example, DS1 requires six bits, four leading zeros, a one, and a trailing zero. The query form of this command returns the serial bit value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

<b>Group</b>	Mask
<b>Syntax</b>	MASK:USER:PATTERNBITS <NR1> MASK:USER:PATTERNBITS?
<b>Related Commands</b>	<a href="#">MASK:USER:PRESAMPBITS</a>
<b>Arguments</b>	<NR1> is an integer that sets the number of bits.
<b>Examples</b>	MASK:USER:PATTERNBITS6 sets the serial bit value of the displayed mask to 6.  MASK:USER:PATTERNBITS? might return :MASK:USER:PATTERNBITS 1, indicating that the serial bit value of the displayed mask is set to 1.

## MASK:USER:PRESAMPBITS

This command sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing. For example, DS1 has four leading zeros. The query form of this command returns the presample bit value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:PRESAMPBITS <NR1>  
MASK:USER:PRESAMPBITS?

**Related Commands** [MASK:USER:PATTERNBITS](#)

**Arguments** <NR1> is an integer that sets the number of bits before the trigger pulse.

**Examples** MASK:USER:PRESAMPBITS4 sets the presample bit value of the displayed mask to 4.

MASK:USER:PRESAMPBITS? might return :MASK:USER:PRESAMPBITS 0, indicating that the presample bit value of the displayed mask is set to 0.

## MASK:USER:RECOrdlength

This command sets or returns the nominal record length for pulse mask testing. The query form of this command returns the record length value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:RECOrdlength <NR1>  
MASK:USER:RECOrdlength?

**Arguments** <NR1> is an integer number that sets the record length.

**Examples** MASK:USER:RECORDLENGTH500 sets the record length value of the displayed mask to 500 points.



`MASK:USER:RECORDLENGTH?` might return `:MASK:USER:RECORDLENGTH 0`, indicating that the record length value of the displayed mask is set to 0.

## **MASK:USER:SEG<m> (No Query Form)**

This command deletes the specified mask segment from the user mask, whether or not the user mask is the current mask.

`m` is an integer that specifies the user mask segment number to delete from the user mask.

**Group** Mask

**Syntax** `MASK:USER:SEG<m> DELEte`

**Arguments** `DELEte` removes the specified segment from the mask.

**Examples** `MASK:USER:SEG3 DELETE` deletes mask segment three from the user mask.

## **MASK:USER:SEG<m>:NR\_Pt? (Query Only)**

This query-only command returns the number of points that make up the specified user mask segment. Each mask point consists of a pair of X-Y coordinates.

`m` is an integer that specifies a user mask segment number.

**Group** Mask

**Syntax** `MASK:USER:SEG<m>:NR_Pt?`

**Returns** `<NR1>` is 0 to 50.

**Examples** `MASK:USER:SEG2:NR_PT?` might return `:MASK:USER:SEG2:NR_PT 5`.

## **MASK:USER:SEG<m>:POINTS**

This command sets or returns the X-Y user coordinates of all points in the specified user mask segment. The set form defines new points in the user mask,

replacing any existing points in the specified user mask segment. It sets or returns the vertices for a particular segment in the selected mask.

m is an integer that specifies the user mask segment number.

**Group** Mask

**Syntax** MASK:USER:SEG<m>:POINTS <NR3>, <NR3> [, <NR3>, <NR3>]  
 MASK:USER:SEG<m>:POINTS?

**Arguments** <NR3> refers to the coordinates of one of the vertices in the user mask. Each pair of numbers represents the horizontal and vertical coordinates of a mask segment vertex. The pairs must be listed in a counterclockwise order. If the vertical or horizontal scale or position is changed after this command and then the query form of this command is issued, the value returned from the instrument will not be the same. If just one pair is input, it is ignored and the user mask segment is marked as undefined. The default is not points in the user mask segment.

**Examples** MASK:USER:SEG2:POINTS -2.3E-9, 44E-3, -2.5E-9, 47E-3, 1.2E-9, 44E-3 defines the points in user mask segment 2.

MASK:USER:SEG2:POINTS? might return :MASK:USER:SEG2:POINTS 0.0000 0.0000, indicating that the X-Y user coordinates for user mask segment 2 in the selected mask is set to 0, 0.

## MASK:USER:TRIGTOSAMP

This command sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position. The query form of this command returns the time value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:TRIGTOSAMP <NR3>  
 MASK:USER:TRIGTOSAMP?

**Related Commands** [MASK:USER:WIDTH](#)

**Arguments** <NR3> is the floating point number that sets the time to the pulse bit sampling position.

**Examples** MASK:USER:TRIGTOSAMP 162E-9 sets the user mask time from the (leading edge) trigger position to the pulse bit sampling position to 162 nanoseconds.

MASK:USER:TRIGTOSAMP? might return :MASK:USER:TRIGTOSAMP 0.0000, indicating that the time value of the displayed mask is set to 0 nanosecond.

## MASK:USER:VOFFSet

This command sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels. The query form of this command returns the offset value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:VOFFSet <NR3>  
MASK:USER:VOFFSet?

**Related Commands** [MASK:USER:VPOS](#), [MASK:USER:VSCALE](#)

**Arguments** <NR3> is a floating point number that sets the nominal vertical offset in volts.

**Examples** MASK:USER:VOFFSET 0.0 sets the user mask nominal vertical offset to 0.0 volts.

MASK:USER:VOFFSET? might return :MASK:USER:VOFFSET 0.0000, indicating that the nominal vertical offset value of the displayed mask is set to 0 volts.

## MASK:USER:VPOS

This command sets or returns the nominal vertical position, control in divisions, used to vertically position the input channels. The query form of this command returns the vertical position value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:VPOS <NR3>  
MASK:USER:VPOS?

**Related Commands** [MASK:USER:VOFFSet](#), [MASK:USER:VSCALE](#)

**Arguments** <NR3> is a floating point number that sets the nominal vertical position control in divisions.

**Examples** MASK:USER:VPOS -0.8 sets the user mask nominal vertical position to -0.8 divisions.

MASK:USER:VPOS? might return :MASK:USER:VPOS 0.0000, indicating that the nominal vertical value of the displayed mask is set to 0 divisions.

## MASK:USER:VSCALE

This command sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels. The query form of this command returns the vertical scale value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:VSCALE <NR3>  
MASK:USER:VSCALE?

**Related Commands** [MASK:USER:VOFFSet](#), [MASK:USER:VPOS](#)

**Arguments** <NR3> is a floating point number that sets the nominal bit width in seconds.

**Examples** MASK:USER:VSCALE200E-3 sets the user mask nominal vertical scale to 200 millivolts per division.

MASK:USER:VSCALE? might return :MASK:USER:VSCALE 1.0000E-3, indicating that the nominal vertical scale of the displayed mask is set to 1 millivolt per division.

## MASK:USER:WIDTH

This command sets or returns the nominal bit width in seconds. The query form of this command returns the bit width value of the displayed mask. The set form of this command only affects the user mask, regardless of the current (displayed) mask.

**Group** Mask

**Syntax** MASK:USER:WIDTH <NR3>  
MASK:USER:WIDTH?

**Related Commands** MASK:USER:TRIGTOSAMP

**Arguments** <NR3> is a floating point number that sets the nominal bit width in seconds. This is the time of one bit of data where bit width = 1 / (data rate of the signal).

**Examples** MASK:USER:WIDTH648E-9 sets the user mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

MASK:USER:WIDTH? might return :MASK:USER:WIDTH 1.0000E-3, indicating that the nominal vertical scale of the displayed mask is set to 1 millivolt per division.

## MATHArbflt<x>:FILEpath

This command or query sets the file path for a file of filter coefficients. Setting a path will read that file and load the filter for ARBFLT<x>, where x can be 1 to 4.

**Group** Math

**Syntax** MATHArbflt<x>:FILEpath <QString>  
MATHArbflt<x>:FILEpath?

**Arguments** <QString> specifies the path to the file of filter coefficients.

**Examples** MATHARBFLT1:FILEPATH "c:/myfilters/20mhz.flr" reads filter coefficients from the file and loads the filter for ARBFLT1.

MATHARBFLT1:FILEPATH? might return :MATHARBFLT1:FILEPATH "C:\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass\_0.bw.flr", indicating that the path to the file of filter coefficients is set to "C:\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass\_0.bw.flr".

## MATHArbflt<x>:READFile (No Query Form)

This command forces a reread of the filter file for each arbitrary filter in use. The <x> specifies the filter and can be 1 to 4.

<b>Group</b>	Math
<b>Syntax</b>	MATHArbflt<x>:READFile FORCE
<b>Arguments</b>	FORCE forces a reread of the filter file.
<b>Examples</b>	MATHARBFLT1:READFILE FORCE forces a reread of the filter file at MATHARBFLT1:FILEPATH location for each arbitrary filter in use.

## MATHVAR? (Query Only)

This command queries all 8 numerical values you can use within math expressions.

<b>Group</b>	Math
<b>Syntax</b>	MATHVAR?
<b>Related Commands</b>	<a href="#">MATHVAR:VAR&lt;x&gt;</a> , <a href="#">MATH&lt;x&gt;:DEFine</a>
<b>Returns</b>	<NR3> are the stored numerical values.
<b>Examples</b>	MATHVAR? returns the values of all variables stored in locations 1 through 8.

## MATHVAR:VAR<x>

This command sets or queries one of 8 different numerical values you can use within math expressions. These values can range from -9.9e37 to 9.9e37; the default values are 0.0. <x> specifies the location, 1 through 8, in which you can store values. Stored math variables can be referenced within math expressions as VAR1, VAR2,...VAR8.

For example, the following command defines MATH1 as the product of Channel 1 and math variable 1: MATH1:DEFINE "CH1 \* VAR1".

<b>Group</b>	Math
<b>Syntax</b>	MATHVAR:VAR<x> <NR3> MATHVAR:VAR<x>?

**Related Commands** [MATHVAR?](#), [MATH<x>:DEFine](#)

**Arguments** <NR3> specifies the numerical value to be stored in location x <1 through 8>.

**Examples** `MATHVAR:VAR3 -2.43E-5` stores the value -2.43e-5 in the third math variable location.  
`MATHVAR:VAR3?` might return `:MATHVAR:VAR3 24.3000E-6` for the expression stored in location 3.

## MATH<x>? (Query Only)

This query-only command returns the definition for the math waveform specified by <x>, which ranges from 1 through 4.

**Group** Math

**Syntax** `MATH<x>?`

**Related Commands** [SElect:<wfm>](#)

**Examples** `MATH1?` might return the following definition of a math waveform: `:MATH1:DEFINE "Ch1-Ch2";UNITSTRING "V";NUMAVG 2;VERTICAL:SCALE 112.0000E-3; POSITION 1.6400;:MATH1:LABEL:NAME "";XPOS 1.0000;YPOS 600.0000E-3;:MATH1:SPECTRAL:MAG DB;PHASE DEGREES;GATEPOS 0.0000;GATEWIDTH 4.0000E-6;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN 625.0000E+6;CENTER 312.5000E+6;RESBW 500.0000E+3;WINDOW GAUSSIAN;SUPPRESS -200.0000;UNWRAP 0;LOCK 0.`

## MATH<x>:DEFine

This command allows you to define new waveforms using mathematical expressions. Sending this command is equivalent to selecting Math Setup from the Math menu, selecting a math waveform (Math 1 through Math 4), and then entering a math expression in the Math<x> box. The query form of this command returns the math definition for the specified math waveform.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants. You can define and display up to four math waveforms simultaneously.

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.

For more information about constructing mathematical expressions, see *Creating and Using Math Waveforms* in the user online help for this instrument.

<b>Group</b>	Math
<b>Syntax</b>	MATH<x>:DEFine <QString> MATH<x>:DEFine?
<b>Related Commands</b>	<a href="#">MATHVAR:VAR&lt;x&gt;</a>
<b>Arguments</b>	<QString> quoted string argument is the mathematical expression that defines the waveform.
<b>Examples</b>	MATH2:DEFINE "CH1+CH2" adds the Channel 1 and Channel 2, storing the results in Math 2 waveform.  MATH1:DEFINE? might return  :MATH1:DEFINE "CH2*REF2" as the expression that defines Math 1 waveform.

## MATH<x>:FILTER:MODE

This command or query sets or returns the filter risetime parameter.

<b>Group</b>	Math
<b>Syntax</b>	MATH<x>:FILTER:MODE {CENTERed SHIFted} MATH<x>:FILTER:MODE?
<b>Arguments</b>	CENTERED sets the value at any point to the average of that point in the source waveform and N points on either side of that point.  SHIFTED sets the value at any point to the average of that point in the source waveform and 2N points before that in the source waveform. This shifts a rising



edge to the right side of the screen. Shifted mode is sometimes called a causal filter since the value at any point is not caused by points after it in time.

**Examples** `MATH1:FILTER:MODE SHIFTED` sets the filter mode to shifted.

`MATH1:FILTER:MODE?` might return `:MATH1:FILTER:MODE CENTERED`, indicating that the filter rise time parameter is set to `CENTERED` mode for Math 1 waveform.

## MATH<x>:FILTer:RISetime

This command or query sets or returns the filter rise time parameter.

**Group** Math

**Syntax** `MATH<x>:FILTer:RISetime <NR3>`  
`MATH<x>:FILTer:RISetime?`

**Arguments** `<NR3>` sets how the filter affects a signal. The bandwidth of the filter is approximately  $0.35 / (\text{filter rise time})$ . For a square wave input, the measurement system rise time of  $\text{Math}(x) = \text{filter}(\text{chx})$  is very close to the filter rise time of  $\text{Math}(x)$ .

**Examples** `MATH1:FILTER:RISETIME 1.7e-3` sets the filter rise time to 1.7 milliseconds.

`MATH1:FILTER:RISETIME?` might return

`:MATH1:FILTER:RISETIME 1.0000E-9`, indicating that the filter rise time parameter is set to 1 nanoseconds for Math 1 waveform.

## MATH<x>:LABel:NAME

This command sets or returns 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`, which ranges in value from 1 through 4. This command is equivalent to selecting Math Setup from the Math menu and entering a label in the Label box.

**Group** Math

**Syntax** `MATH<x>:LABel:NAME <QString>`  
`MATH<x>:LABel:NAME?`

**Arguments** <Qstring> specifies the label to annotate the math waveform.

**Examples** MATH2:LABEL:NAME "PROBE POINT7" assigns "Probe point7" Math 2 waveform.  
MATH2:LABEL:NAME? might return :MATH2:LABEL:NAME "Probe point7", indicating that Probe point 7 is the label for Math 2 waveform.

## MATH<x>:LABel:XPOS

This command sets or queries the X screen offset at which the label attached to a math waveform is displayed, relative to the left edge of the screen. Channels are specified by x, which ranges from 1 through 4. This command is equivalent to selecting Math Label from the Math menu and entering a value in the X Position box.

**Group** Math

**Syntax** MATH<x>:LABel:XPOS <NR1>  
MATH<x>:LABel:XPOS?

**Related Commands** [MATH<x>:LABel:YPOS](#)

**Arguments** <NR1> is the location (in divisions) where the label for the selected math waveform is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 to 10.

**Examples** MATH2:LABEL:XPOS 5 moves the waveform label for the Math 2 waveform (on a four-channel instrument) so that it begins 5 divisions to the right of the left edge of the screen.  
MATH2:LABEL:XPOS? might return :MATH2:LABEL:XPOS 2.5, indicating that the waveform label for the Math 2 waveform is currently 2.5 divisions to the right of the left edge of the screen.

## MATH<x>:LABel:YPOS

This command sets or queries the Y screen offset at which the label attached to a math waveform is displayed, relative to the waveform handle. The Math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Math Label from the Math menu and entering a value in the Y Position box.

<b>Group</b>	Math
<b>Syntax</b>	MATH<x>:LABe1:YPOS <NR1> MATH<x>:LABe1:YPOS?
<b>Related Commands</b>	<a href="#">MATH&lt;x&gt;:LABel:XPOS</a>
<b>Arguments</b>	<NR1> is the location (in divisions) where the label for the selected math waveform is displayed, relative to the waveform handle. Arguments should range from 10 to -10.
<b>Examples</b>	MATH2:LABEL:YPOS -2.5 moves the waveform label for the Math 2 waveform to 2.5 divisions below the waveform handle.  MATH2:LABEL:YPOS? might return :MATH2:LABEL:YPOS 0, indicating that the waveform label for the Math 2 waveform is currently located at the waveform handle.

## MATH<x>:NUMAVg

This command sets or returns the acquisition number at which the averaging algorithm will begin exponential averaging. Prior to that acquisition number, the algorithm uses stable averaging. This has no effect unless the AVG() function is used in the specified math expression. If so, it affects all AVG() functions in this math expression. The Math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Set Math Averages from the Math menu and then entering an averaging value for the math waveform.

<b>Group</b>	Math
<b>Syntax</b>	MATH<x>:NUMAVg <NR1> MATH<x>:NUMAVg?
<b>Related Commands</b>	<a href="#">ACQuire:NUMAVg</a>
<b>Arguments</b>	<NR1> specifies the number of acquisitions over which exponential averaging is performed.

**Examples**    `MATH2:NUMAVG 10` averages the Math 2 waveform ten times.

`MATH2:NUMAVG?` might return `:MATH2:NUMAVG 10`, indicating that ten Math 2 waveforms are averaged before a single acquisition occurs.

## MATH<x>:SPECTral? (Query Only)

This query-only command returns the current spectral setups for the specified math waveform. The Math waveform is specified by *x*, which ranges from 1 through 4. This command is equivalent to selecting Spectral Setup from the Math menu and viewing the current spectral setup values.

**Group**    Math

**Syntax**    `MATH<x>:SPECTral?`

**Examples**    `MATH1:SPECTRAL?` might return the spectral setup values for MATH1 as follows: `:MATH1:SPECTRAL:MAG DB; PHASE DEGREES; GATEPOS 0.0000E+00; GATEWIDTH 1.9996E-06; REFLEVEL 4.4587+01; REFLEVELOFFSET2.2360E-01; SPAN 1.2500E+09; CENTER 6.2500E+08; RESBW 1.0002E=06; WINDOW GAUSSIAN; SUPPRESS -3.5000E+01; UNWRAP 0;LOCK 0.`

## MATH<x>:SPECTral:CENTER

This command specifies or returns the center frequency of the spectral analyzer output data span for the specified math waveform. The Math waveform is specified by *x*, which ranges from 1 through 4. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Center Freq value.

**Group**    Math

**Syntax**    `MATH<x>:SPECTral:CENTER <NR3>`  
`MATH<x>:SPECTral:CENTER?`

**Related Commands**    [MATH<x>:SPECTral:SPAN](#), [HORizontal:MODE:SAMPLERate](#)

**Arguments**    `<NR3>` is the desired frequency of the spectral analyzer output data span in hertz.

**Examples**    `MATH3:SPECTRAL:CENTER 10.09E6` sets the center frequency to the closest value it can attain to 10.09 MHz.

`MATH2:SPECTRAL:CENTER?` might return `:MATH2:SPECTRAL:CENTER 10.0900E+06`, indicating that the center frequency is currently set at 10.09 MHz.

## MATH<x>:SPECTral:GATEPOS

This command sets or returns the position of the center of the gate, which is used as the data input to the spectral analyzer for the specified math waveform. The math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Gate Pos value.

**Group**    Math

**Syntax**    `MATH<x>:SPECTral:GATEPOS <NR3>`  
`MATH<x>:SPECTral:GATEPOS?`

**Related Commands**    [MATH<x>:SPECTral:GATEWIDTH](#)

**Arguments**    <NR3> is the gate position. Units are represented in seconds, with respect to trigger position.

**Examples**    `MATH1:SPECTRAL:GATEPOS -2e8` specifies the position of the edge of the gate used as the data input to the spectral analyzer. The `-2e-8` specifies that the center of the gate is  $2e-8$  seconds before (to the left of) the trigger.

`MATH2:SPECTRAL:GATEPOS?` might return `:MATH2:SPECTRAL:GATEPOS 0.0000E+00`, indicating that the gate position is set to zero (the trigger position).

## MATH<x>:SPECTral:GATEWIDTH

This command sets or returns the gate width input, in seconds, to the spectral analyzer for the specified math waveform. The math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Spectral Setup from the Math menu and entering a duration value in the Gate Dur box.

**Group**    Math

**Syntax**     MATH<x>:SPECTral:GATEWIDTH <NR3>  
MATH<x>:SPECTral:GATEWIDTH?

**Related Commands**     [MATH<x>:SPECTral:GATEPOS](#)

**Arguments**     <NR3> is the time across the 10-division screen in seconds.

**Examples**     MATH1:SPECTRAL:GATEWIDTH 1.0E-3 sets the gate width input to the spectral analyzer at 1 ms.

MATH3:SPECTRAL:GATEWIDTH? might return :MATH3:SPECTRAL:GATEWIDTH 1.0000E-03, indicating that the gate width to the spectral analyzer is set to 1 ms.

## MATH<x>:SPECTral:LOCK

This command locks menus for two or more math waveforms together as a group. The query form of this command returns an ON (1) or OFF (0), indicating whether spectral locking is turned on. This command is equal to selecting Spectral Setup from the Math menu, choosing the Control tab and then clicking the Time/Track Frequency Domain Controls button associated with the math waveforms that you want to lock.

**Table 2-44: Math<x> Lock Combinations**

Math1	Math2	Math3	Locked Math Waveforms
Off	Off	Off	None
Off	Off	On	Math3 and Math4 locked
Off	On	Off	Math2 and Math3 locked
Off	On	On	Math2, Math3, and Math4 locked
On	Off	Off	Math1 and Math2 locked
On	Off	On	Math1 and Math2 locked, Math3 and Math4 locked
On	On	Off	Math1, Math2, and Math3 locked
On	On	On	Math1, Math2, Math3, and Math4 locked

---

**NOTE.** Executing *MATH4:SPECTral:LOCK* via the GPIB interface has no effect since there is no Math5 to which it can be locked.

---

**Group**     Math

**Syntax** MATH<x>:SPECTral:LOCK {ON|OFF|<NR1>}  
MATH<x>:SPECTral:LOCK?

**Arguments** ON turns on the parameter lock for the specified math waveform.  
OFF turns off the parameter lock for the specified math waveform.  
<NR1> = 0 disables the parameter lock for the specified math waveform; any other value enables the parameter lock.

**Examples** MATH1:SPECTRAL:LOCK ON turns on the parameter lock for Math 1, which causes the parameters for Math 1 and Math 2 waveforms to be locked together.  
MATH1:SPECTRAL:LOCK? might return :MATH1:SPECTRAL:LOCK 0, indicating that the parameter lock for Math 1 waveform is turned off.

## MATH<x>:SPECTral:MAG

This command sets or returns the units of the SpectralMag function in the specified math definition string. The Math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Spectral Mag from the Math menu and then entering the units that you want in the Scale box, or selecting Basic from the Math menu and then clicking the desired Scale button.

**Group** Math

**Syntax** MATH<x>:SPECTral:MAG {LINEAR|DB|DBM}  
MATH<x>:SPECTral:MAG?

**Arguments** LINEAR sets the SpectralMag units to linear.  
DB sets the SpectralMag units to decibels.  
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  $\Omega$ .

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

## MATH<x>:SPECTral:PHASE

This command sets or returns the units of a SpectralPhase function in the specified math definition string. The Math waveform is specified by x, which ranges from 1 through 4. This command is equal to selecting Spectral Phase from the Math menu, selecting the Advanced button, selecting the Vert Axis tab, and then clicking the desired Scale button.

**Group** Math

**Syntax** MATH<x>:SPECTral:PHASE {DEGREES|RADIANS|GROUPDelay}  
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<x>:SPECTral:REFLevel

This command specifies the vertical position of the specified spectral math waveform on the display screen. The numerical value represents the position at the top of the display graticule. The Math waveform is specified by x, which ranges from 1 through 4. This command is equal to selecting Spectral Setup from the Math menu, choosing the Mag tab and then entering a value in the Reference Level box.

**Group** Math

**Syntax** MATH<x>:SPECTral:REFLevel <NR3>  
MATH<x>:SPECTral:REFLevel?

**Related Commands** [MATH<x>:SPECTral:REFLEVELOffset](#)



- Arguments** <NR3> is the value that represents the top of the display screen graticule. The range depends on the units and both the MATH<x>:VERTICAL:SCALE and MATH<x>:VERTICAL:POSITION settings.
- Examples** MATH1:SPECTRAL:REFLEVEL -10 sets the top of the display screen to be a reference level of -10 dB. If the vertical scale is LINEAR, then the vertical units will be the same as the input waveform.
- MATH1:SPECTRAL:REFLEVEL? might return :MATH1:SPECTRAL:REFLEVEL 2.0000E+01, indicating that the top of the display screen is set to a reference level of 20 dB.

## MATH<x>:SPECTRAL:REFLEVELOffset

This command sets or returns the spectral level offset used for calculating the dB value for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. Changing the reference level offset causes the spectral waveform to move vertically, with respect to zero dB. This command is equal to selecting Spectral Setup from the Math menu, choosing the Mag tab and then entering a value in the Reference Level Offset box.

**Group** Math

**Syntax** MATH<x>:SPECTRAL:REFLEVELOffset {DBM|<NR3>}  
MATH<x>:SPECTRAL:REFLEVELOffset?

**Related Commands** [MATH<x>:SPECTRAL:REFLevel](#)

**Arguments** DBM specifies the reference level used for calculation to be equivalent to 1 mW into 50 Ω (Zero dB will occur at this level).

<NR3> specifies the reference level used for calculation of the decibel value when the output units are Log.

**Examples** MATH1:SPECTRAL:REFLEVELOFFSET 0.5 sets the reference level for the Log calculation for decibel to be 0.5.  $dB = A \times \text{Log}(y/\langle NR3 \rangle)$  where A is 10 if the input units are watts and A is 20 if the input units are not watts.

MATH1:SPECTRAL:REFLEVELOFFSET DBM sets the decibel reference to be equivalent to 1 mW into 50 Ω. The reference level numerical value will depend on the input units. If the units are volts, the value is set to 0.2236 V; if the units are amperes, the value is set to 40 μA; if the units are watts, the value is set to 0.001 W.

MATH1:SPECTRAL:REFLEVELOFFSET? might return  
:MATH1:SPECTRAL:REFLEVELOFFSET 2.23360E-01, indicating that the  
spectral reference level offset is 223.6 mV.

## MATH<x>:SPECTral:RESBw

This command sets or returns the resolution bandwidth of the spectral analyzer for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a value in the Res BW box.

**Group** Math

**Syntax** MATH<x>:SPECTral:RESBw <NR3>  
MATH<x>:SPECTral:RESBw?

**Related Commands** [MATH<x>:SPECTral:GATEWIDTH](#), [MATH<x>:SPECTral:CENTER](#),  
[MATH<x>:SPECTral:SPAN](#), [MATH<x>:SPECTral:WINDOW](#)

**Arguments** <NR3> is the desired resolution bandwidth value. Units are represented in Hertz.

**Examples** MATH1:SPECTRAL:RESBw 250E3 sets the resolution bandwidth to the attainable  
value that is close to 250 kHz.

MATH1:SPECTRAL:RESBw? might return :MATH1:SPECTRAL:RESBw  
1.0002E+06, indicating the actual resolution bandwidth value obtained from  
the spectral analyzer.

## MATH<x>:SPECTral:SPAN

This command sets the ceiling of the frequency span to a value that is closest to the specified value. The query form of this command returns the current span value for specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. This command is equal to selecting Spectral Setup from the Math menu and then entering a value in the Freq Span box.

**Group** Math

**Syntax** MATH<x>:SPECTral:SPAN {<NR3>|FUL1}  
MATH<x>:SPECTral:SPAN?

**Related Commands** [MATH<x>:SPECTral:CENTER](#)

**Arguments** <NR3> specifies the frequency span of the output data vector from the spectral analyzer.

FULL sets the top of the span to 1/2 the sample rate and sets the center frequency to 1/2 the span.

**Examples** MATH1:SPECTRAL:SPAN FULL sets the top of the span to 1/2 the sample rate and sets the center frequency to 1/2 the span.

MATH1:SPECTRAL:SPAN 2.56E6 sets the top of the span to a value that is closest to 2.56 MHz.

MATH1:SPECTRAL:SPAN? might return :MATH1:SPECTRAL:SPAN 1.2500E+09, indicating the actual span value obtained by the spectral analyzer.

## MATH<x>:SPECTral:SUPPpress

This command sets or returns the phase suppression threshold for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then entering a value in the Suppression Threshold box. This command affects only Spectral Phase waveforms.

**Group** Math

**Syntax** MATH<x>:SPECTral:SUPPpress <NR3>  
MATH<x>:SPECTral:SUPPpress?

**Related Commands** [MATH<x>:SPECTral:UNWRap](#)

**Arguments** <NR3> is the magnitude level that data with magnitude values below this value are displayed as zero phase.

**Examples** MATH1:SPECTRAL:SUPPRESS -62 specifies that any data with magnitude values less than -62 dB will have their phase output set to zero.

MATH1:SPECTRAL:SUPPRESS? might return :MATH1:SPECTRAL:SUPPRESS -3.5000E+01, indicating that the phase suppression threshold is currently set to -35 dB.

## MATH<x>:SPECTral:UNWRap

This command sets or returns whether phase unwrap of the spectral analyzer output data is enabled for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then clicking the Unwrap button. This command affects only Spectral Phase waveforms.

**Group** Math

**Syntax** MATH<x>:SPECTral:UNWRap {ON|OFF|<NR1>}  
MATH<x>:SPECTral:UNWRap?

**Related Commands** [MATH<x>:SPECTral:SUPPress](#)

**Arguments** ON enables phase unwrap.  
OFF disables phase wrap.  
<NR1> = 0 disables phase wrap; any other value enables phase wrap.

**Examples** MATH1:SPECTRAL:UNWRAP ON enables phase wrap of the spectral analyzer output data.  
MATH1:SPECTRAL:UNWRAP? might return :MATH1:SPECTRAL:UNWRAP 0, indicating that the phase unwrap of the spectral analyzer output data is disabled.

## MATH<x>:SPECTral:WINDow

This command sets or returns the window function used to multiply the spectral analyzer input data for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. 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. This command is equal to selecting Spectral Setup from the Math menu, and choosing from the Window Type drop-down list.

Following is a list of arguments that specify the window function used to multiply the spectral analyzer input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth). For additional information about spectral windows, see *Selecting a Spectral Window* in the online help for this instrument.

**Group** Math

**Syntax** MATH<x>:SPECTral:WINDOW  
 {RECTANGular|HAMMING|HANNing|KAISERBessel|BLACKMANHarris|FLATTOP2|GAUSSian|TEKEXPONENTIAL|TEKEXPONENTIAL?}  
 MATH<x>:SPECTral:WINDOW?

**Related Commands** [MATH<x>:SPECTral:RESBw](#)

**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.  
 KAISERBessel window function is based on a cosine series.  
 BLACKMANHarris window function is based on a cosine series.  
 FLATTOP2 window function is a cosine series window with a flattened frequency response lobe.  
 GAUSSian window function has the best localization characteristics in the joint time/frequency plane.  
 TEKEXPONENTIAL window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

**Examples** MATH2:SPECTRAL:WINDOW HANNING applies a Hanning window to the spectral analyzer input data.  
 MATH2:SPECTRAL:WINDOW? might return :MATH2:SPECTRAL:WINDOW  
 TEKEXPONENTIAL, indicating that the window function used to multiply the spectral analyzer input data is the Tek exponential function.

## MATH<x>:THRESHold

This command sets or queries the threshold for the math waveform specified by x, which can be 1 through 4.

**Group** Math

**Syntax** MATH<x>:THRESHold <NR3>  
 MATH<x>:THRESHold?

**Related Commands** [CH<x>:THRESHold](#), [D<x>:THRESHold](#)

**Arguments** <NR3> specifies the math threshold, in volts.

**Examples** MATH1:THRESHOLD 1 sets the threshold value of the MATH1 to 1 volt.  
MATH1:THRESHOLD? might return MATH1:THRESHOLD 500.0000E-3, indicating that the threshold value set for the MATH1 is 0.5 volts.

## MATH<x>:UNITString

This command sets or queries the string to use for units for the math waveform specified by x, which can be 1 through 4. This command will override the default unit string with the one that you specify.

**Group** Math

**Syntax** MATH<x>:UNITString <QString>  
MATH<x>:UNITString?

**Arguments** <QString> quoted string argument is the units to be used for the specified math waveform.

**Examples** MATH2:UNITString "deg" specifies that the unit of the Math 2 waveform operation will be called deg.  
MATH1:UNITSTRING? might return :MATH1:UNITSTRING "deg" as the unit for the Math 1 waveform operation.

## MATH<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Math waveform. The Math waveform is specified by x, which ranges from 1 through 4. The position value is usually applied to the signal before it is digitized. The highest three units/div scale ranges of a given math are implemented by changing the way the acquired data is displayed. When the instrument is operating in any of these highest three scale ranges, the position control operates only on the signal after it is digitized. Note that if a signal that exceeds the range of the digitizer in one of these three scale ranges is repositioned, the displayed waveform will contain clipped values on-screen. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Pos value or adjusting the front panel Vertical POSITION knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal to the present offset setting for that reference, are

displayed. For example, if the position for Math 3 is set to 2.0 and the offset is set to 3.0, then the input signals equal to 3.0 are displayed 2.0 divisions above the center of the screen.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max cover 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

**Group** Math

**Syntax** MATH<x>:VERTical:POSition <NR3>  
MATH<x>:VERTical:POSition?

**Related Commands** [CH<x>:POSition](#), [REF<x>:VERTical:POSition](#)

**Arguments** <NR3> is the desired position control in divisions from the center graticule.

**Examples** MATH2:VERTICAL:POSITION 1.3E+00 positions the Math 2 waveform input signal 1.3 divisions higher than a position of 0.

MATH1:VERTICAL:POSITION? might return :MATH1:VERTICAL:POSITION -1.3000E+00 indicating that the current position of Math 1 waveform is 1.3 divisions below the center graticule.

## MATH<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Scale value or adjusting the front panel Vertical SCALE knob.

Each waveform has its own vertical scale parameter. For a signal with constant amplitude, increasing the scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms. For reference and math waveforms, the scale setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max covers 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

**Group** Math

**Syntax** MATH<x>:VERTICAL:SCALE <NR3>  
MATH<x>:VERTICAL:SCALE?

**Related Commands** CH<x>:SCALE, REF<x>:VERTICAL:SCALE

**Arguments** <NR3> is the scale, in volts, amps or watts per division. The range is from 100.0E-36 through 100.0E+36.

**Examples** MATH4:VERTICAL:SCALE 100E-03 sets the Math 4 waveform scale to 100 mV per division.  
MATH2:VERTICAL:SCALE? might return :MATH2:VERTICAL:SCALE 1.0000E+00, indicating that the current scale setting of Math 2 waveform is 1 V per division.

## MEASUREMENT? (Query Only)

This query-only command returns all measurement parameters in the following order: MEAS1, MEAS2, MEAS3, MEAS4, MEAS5, MEAS6, MEAS7, MEAS8, and IMMED.

**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:ANNOTATION:IMMEDSTATE

This command sets or queries the state of immediate measurement annotation.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:ANNOTATION:IMMEDSTATE {<NR1> OFF ON} MEASUREMENT:ANNOTATION:IMMEDSTATE?
<b>Arguments</b>	NR1 = 0 disables immediate state measurement annotation, any other value enables immediate state measurements.  OFF disables annotation on immediate state measurements.  ON enables annotation on immediate state measurements.
<b>Examples</b>	MEASUREMENT:ANNOTATION:IMMEDSTATE ON enables annotation of immediate measurements.  MEASUREMENT:ANNOTATION:IMMEDSTATE? might return :MEASUREMENT:ANNOTATION:IMMEDSTATE 0, indicating that annotation of immediate measurements is off.

## MEASUrement:ANNOtation:NUMX? (Query Only)

This query-only command returns the number of horizontal measurement annotations currently being displayed.

**Group** Measurement

**Syntax** MEASUrement:ANNOtation:NUMX?

**Examples** MEASUREMENT:ANNOTATION:NUMX? might return  
MEASUREMENT:ANNOTATION:NUMX 2, indicating that there are currently 2 horizontal lines drawn on the graticule showing where the measurement specified by MEASUREMENT:ANNOTATION:STATE is being performed.

## MEASUrement:ANNOtation:NUMY? (Query Only)

This query-only command returns the number of vertical measurement annotations currently being displayed.

**Group** Measurement

**Syntax** MEASUrement:ANNOtation:NUMY?

**Examples** MEASUREMENT:ANNOTATION:NUMY? might return  
MEASUREMENT:ANNOTATION:NUMY 2, indicating that there are currently 2 vertical lines drawn on the graticule showing where the measurement specified by MEASUREMENT:ANNOTATION:STATE is being performed.

## MEASUrement:ANNOtation:STATE

This command sets or returns the state of visible measurement annotations.

**Group** Measurement

**Syntax** MEASUrement:ANNOtation:STATE  
{OFF|MEAS1|MEAS2|MEAS3|MEAS4|MEAS5|MEAS6|MEAS7|MEAS8}  
MEASUrement:ANNOtation:STATE?

- Arguments** OFF turns off visible measurement annotations.
- MEAS<x> turns on the display of visible measurement annotations for measurement <x>, where <x> can be 1, 2, 3, 4, 5, 6, 7, or 8. There must be an active measurement before you can activate an annotation for a specified measurement.
- Examples** MEASUREMENT:ANNOTATION:STATE MEAS2 turns on the display of visible measurement annotations for measurement 2.
- MEASUREMENT:ANNOTATION:STATE? might return MEASUREMENT:ANNOTATION:STATE OFF, indicating that no measurement annotations are active.

## MEASUREMENT:ANNOTATION:TYPE

This command sets or queries the type of annotations being used. This command is similar to setting the annotation type in the Measure > Annotation menu.

- Group** Measurement
- Syntax** MEASUREMENT:ANNOTATION:TYPE {STANDARD|DETAILED}  
MEASUREMENT:ANNOTATION:TYPE?
- Arguments** STANDARD sets the annotation type to STANDARD.
- DETAILED sets the annotation type to DETAILED, so that more detailed annotations are displayed.
- Examples** MEASUREMENT:ANNOTATION:TYPE DETAILED sets the measurement annotation type to detailed.
- MEASUREMENT:ANNOTATION:TYPE? might return :MEASUREMENT:ANNOTATION:TYPE STANDARD, indicating that standard annotations will be displayed.

## MEASUREMENT:ANNOTATION:X<x>? (Query Only)

This query-only command returns the value of the specified X annotation. X annotations are numbered, in general, from top to bottom on the display. The value of <x> can vary from 1 to 6.

- Group** Measurement

**Syntax** MEASUREMENT:ANNOTATION:X<x>?

**Examples** MEASUREMENT:ANNOTATION:X1? might return  
:MEASUREMENT:ANNOTATION:X1 277.2250E-12 , indicating that annotation  
X1 is 277 ps to the right of the trigger.

## MEASUREMENT:ANNOTATION:Y<x>? (Query Only)

This query-only command returns the value of the specified Y annotation. Y annotations are numbered, in general, from left to right on the display. The value of <x> can vary from 1 to 6.

**Group** Measurement

**Syntax** MEASUREMENT:ANNOTATION:Y<x>?

**Examples** MEASUREMENT:ANNOTATION:Y1 might return  
:MEASUREMENT:ANNOTATION:X1 305.6406E-3 , , indicating that annotation  
Y1 is 305 mV above the channel reference.

## MEASUREMENT:GATING

This command specifies or returns the measurement gating setting. This command is equivalent to selecting Gating from the Measure menu and then clicking the desired Measurement Gating setting.

**Group** Measurement

**Syntax** MEASUREMENT:GATING {ON|OFF|<NR1>|ZOOM<x>|CURSOR}  
MEASUREMENT:GATING?

**Arguments** ON turns on measurement gating.  
OFF turns off measurement gating.  
<NR1> = 0 turns off measurement gating; any other value turns on measurement gating.  
ZOOM<x> turns on gating, using the left and right edges of the zoom box. <x> specifies the zoom window, which ranges from 1 through 4.

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

**Examples** MEASUREMENT:GATING ON turns on measurement gating.

MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

## MEASUREMENT:IMMED? (Query Only)

This query-only command returns all immediate measurement setup parameters.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED?

**Examples** MEASUREMENT:IMMED? might return :MEASUREMENT:IMMED:TYPE AMPLITUDE; 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:IMMED:DELAY? (Query Only)

This query-only command returns information about the immediate delay measurement. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab and then clicking the Delay button.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:DELAY?

**Examples** MEASUREMENT:IMMED:DELAY? might return :MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS.

## MEASUrement:IMMed:DELAy:DIREction

This command sets or returns the starting point and direction that determines the delay “to” edge when taking an immediate delay measurement. Use the [MEASUrement:IMMed:SOUrce<x>](#) command to specify the delay “to” waveform. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Search Direction setting.

**Group** Measurement

**Syntax** MEASUrement:IMMed:DELAy:DIREction {BACKwards|FORwards}  
 MEASUrement:IMMed:DELAy:DIREction?

**Related Commands** [MEASUrement:IMMed:SOUrce<x>](#)

**Arguments** BACKwards starts the search at the end of the waveform and looks for the last rising or falling edge in the waveform.  
 FORwards starts the search at the beginning of the waveform and looks for the first rising or falling edge in the waveform.

**Examples** MEASUREMENT:IMMED:DELAY:DIRECTIONFORWARDS starts searching from the beginning of the waveform record and looks for the first rising or falling edge.  
 MEASUREMENT:IMMED:DELAY:DIRECTION? might return  
 :MEASUREMENT:IMMED:DELAY:DIRECTION BACKWARDS, indicating that searching begins at the end of the waveform record and looks for the last rising or falling edge.

## MEASUrement:IMMed:DELAy:EDGE[1]

This command sets or queries the slope of the edge that is used for the delay “from” waveform when taking an immediate delay measurement. Use the [MEASUrement:IMMed:SOUrce<x>](#) command to specify the waveform. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Delay Edge1 setting.

---

**NOTE.** *If you do not specify a numerical suffix, the source is assumed to be SOURCE 1.*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:DELAY:EDGE[1] {FALL RISe} MEASUREMENT:IMMED:DELAY:EDGE[1]?
<b>Related Commands</b>	<a href="#">MEASUREMENT:IMMED:SOURCE&lt;x&gt;</a>
<b>Arguments</b>	FALL specifies the falling edge. RISe specifies the rising edge.
<b>Examples</b>	MEASUREMENT:IMMED:DELAY:EDGE[1] 1 RISe specifies that the rising edge be used for the immediate delay measurement.  MEASUREMENT:IMMED:DELAY:EDGE[1]? might return :MEASUREMENT:IMMED:DELAY:EDGE1 FALL, indicating that the falling edge or negative edge of the waveform is used for the immediate delay measurement.

## MEASUREMENT:IMMED:DELAY:EDGE2

This command sets or queries the slope of the edge that is used for the delay “to” waveform when taking an immediate delay measurement. Use the [MEASUREMENT:IMMED:SOURCE<x>](#) command to specify the waveform. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Delay Edge2 setting.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:DELAY:EDGE2 {FALL RISe} MEASUREMENT:IMMED:DELAY:EDGE2?
<b>Related Commands</b>	<a href="#">MEASUREMENT:IMMED:SOURCE&lt;x&gt;</a>
<b>Arguments</b>	FALL specifies the falling edge. RISe specifies the rising edge.
<b>Examples</b>	MEASUREMENT:IMMED:DELAY:EDGE2RISe specifies that the rising edge be used for the immediate delay measurement.

MEASUREMENT:IMMED:DELAY:EDGE2? might return  
 :MEASUREMENT:IMMED:DELAY:EDGE2 FALL, indicating that the falling edge or  
 negative edge of the waveform is used for the immediate delay measurement.

## MEASUREMENT:IMMED:METHod

This command specifies or queries the method used to calculate the 0% and 100% reference level for immediate measurements.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:METHod {HISTogram|MINMax|MEAN}  
 MEASUREMENT:IMMED:METHod?

**Arguments** HISTogram sets the high and low waveform levels statistically using a histogram algorithm.

MINMax sets the high and low waveform levels to MAX and MIN, respectively.

MEAN sets the high and low waveform levels to their mean.

**Examples** MEASUREMENT:IMMED:METHod HISTOGRAM specifies that the high and low reference levels are set statistically.

MEASUREMENT:IMMED:METHod? might return :MEASUREMENT:IMMED:METHod  
 MINMAX, indicating that the reference levels are set to MIN and MAX for  
 immediate measurements.

## MEASUREMENT:IMMED:NOISe

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined.

**Conditions** This command is only valid if the instrument has Option MTM is installed.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:NOISe {HIGH|LOW}  
 MEASUREMENT:IMMED:NOISe?



- Arguments** HIGH argument causes the measurement for noise to be taken at the high level of the waveform.
- LOW argument causes the measurement for noise to be taken at the low level of the waveform.
- Examples** MEASUREMENT:IMMED:NOISEHIGH causes the noise measurement to be made at the high level of the waveform.
- MEASUREMENT:IMMED:NOISE? might return :MEASUREMENT:IMMED:NOISE LOW, indicating that the noise measurement will be made at the low level of the waveform.

## MEASUREMENT:IMMED:REFLEVEL? (Query Only)

This query-only command returns the reference level settings for the immediate measurement. It returns them in the following order: ABSOLUTE and then PERCENT for individual user measurements.

- Group** Measurement
- Syntax** MEASUREMENT:IMMED:REFLEVEL?
- Examples** MEASUREMENT:IMMED:REFLEVEL? might return :MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90 indicating that the reference level is set to 90% of HIGH.

## MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH

This command sets or queries the high reference level, and is the upper reference level when [MEASUREMENT:IMMED:REFLEVEL:METHOD](#) is set to Absolute. Note that this command affects the results of rise and fall measurements.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

- Group** Measurement
- Syntax** MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH <NR3>  
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH?

**Related Commands**    [MEASUREMENT:IMMED:REFLEVEL:METHOD](#)

**Arguments**    <NR3> is the high reference level, in volts. The default is 0.0 V.

**Examples**    MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH1.71 sets the high reference level to 1.71 V.

MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH? might return  
:MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00, indicating that the absolute high reference level is set to 1.71 V.

## MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW

This command sets or queries the low reference level, and is the zero percent level when [MEASUREMENT:IMMED:REFLEVEL:METHOD](#) is set to Absolute. Note that this command affects the results of rise and fall measurements.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

**Group**    Measurement

**Syntax**    MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW <NR3>  
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW?

**Related Commands**    [MEASUREMENT:IMMED:REFLEVEL:METHOD](#)

**Arguments**    <NR3> is the low reference level, in volts. The default is 0.0 V.

**Examples**    MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW0.0 sets the low reference level to 0.0 V.

MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW? might return  
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW 0.0000E+00, indicating that the absolute low reference level is set to 0.0 V.

## MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID<x>

This command sets or queries the mid reference level, and is the 50% reference level when [MEASUREMENT:IMMED:REFLEVEL:METHOD](#) is set to Absolute. Note

that this command affects the results of period, frequency, delay, and all cyclic measurements. Note that this command affects the results of delay measurements.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID<x> <NR3> MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID<x>?
<b>Related Commands</b>	<a href="#">MEASUREMENT:IMMED:REFLEVEL:METHOD</a>
<b>Arguments</b>	<NR3> is the mid1 (the “from” waveform when taking a delay measurement) or mid2 (the “to” waveform when taking a delay measurement) reference level, in volts. The default is 0.0 V.
<b>Examples</b>	MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2 0.5 sets the mid reference level for the delay waveform to 0.5 V.  MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2? might return :MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2 0.5000E+00, indicating that the absolute mid2 reference level is set to 0.5 V.

## MEASUREMENT:IMMED:REFLEVEL:METHOD

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired reference level from the Units group box.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:REFLEVEL:METHOD {ABSOLUTE PERCENT} MEASUREMENT:IMMED:REFLEVEL:METHOD?
<b>Arguments</b>	ABSOLUTE specifies that the reference levels are set explicitly using the MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE commands. This method is useful when precise values are required. For instance, when designing to published interface specifications, such as RS-232-C.

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:IMMED:REFLEVEL:PERCENT commands.

**Examples** MEASUREMENT:IMMED:REFLEVEL:METHoD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.

MEASUREMENT:IMMED:REFLEVEL:METHoD? might return :MEASUREMENT:IMMED:REFLEVEL:METHoD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

## MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the high reference level when MEASUREMENT:IMMED:REFLEVEL:METHoD is set to Percent. Note that this command affects the results of rise and fall measurements.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH <NR1>  
MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH?

**Related Commands** MEASUREMENT:IMMED:REFLEVEL:METHoD

**Arguments** <NR1> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

**Examples** MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 95 sets the high reference level to 95% of HIGH.

MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH? might return :MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90, indicating that the percentage high reference level is set to 90% of HIGH.

## MEASUrement:IMMed:REFLevel:PERCent:LOW

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the low reference level when [MEASUrement:IMMed:REFLevel:METhod](#) is set to Percent. Note that this command affects the results of rise and fall measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Percentage Low Ref value.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:IMMed:REFLevel:PERCent:LOW <NR1>  
MEASUrement:IMMed:REFLevel:PERCent:LOW?

**Related Commands** [MEASUrement:IMMed:REFLevel:METhod](#)

**Arguments** <NR1> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

**Examples** MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW 15 sets the low reference level to 15% of HIGH.

MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW? might return  
:MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW 10, indicating that the percentage low reference level is set to 10% of HIGH.

## MEASUrement:IMMed:REFLevel:PERCent:MID<x>

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when [MEASUrement:IMMed:REFLevel:METhod](#) is set to Percent. Mid1 is for the first waveform specified, and Mid2 is for the second waveform specified. Note that this command affects the results of delay measurements.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

**Group** Measurement

**Syntax**      `MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID<x> <NR1>`  
`MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID<x>?`

**Related Commands**      [MEASUREMENT:IMMED:REFLEVEL:METHOD](#)

**Arguments**      <NR1> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Examples**      `MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2 40` sets the mid2 reference level to 40% of HIGH.

`MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2?` might return  
`:MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2 45`, indicating that the percentage mid2 reference level is set to 45% of HIGH.

## MEASUREMENT:IMMED:SOURCE<x>

This command sets or queries the source for phase or delay immediate measurements. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Source1 (From) setting or Source2 (To) setting.

Tip: Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.

**Group**      Measurement

**Syntax**      `MEASUREMENT:IMMED:SOURCE<x> {CH<x>|MATH<y>|REF<x>|HISTogram}`  
`MEASUREMENT:IMMED:SOURCE<x>?`

**Arguments**      `CH<x>` is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4.

`MATH<y>` is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4.

`REF<x>` is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4.

`HISTogram` indicates histogram as the object to be measured. `HISTogram` not allowed on `SOURCE2`.

**Examples** MEASUREMENT:IMMED:SOURCE2 REF3 sets the waveform in reference memory location 3 as the delay “to” source when making delay measurements.

MEASUREMENT:IMMED:SOURCE2? might return  
:MEASUREMENT:IMMED:SOURCE2 MATH1, indicating that Math1 is the immediate measurement source.

## MEASUREMENT:IMMED:SOURCE<x>:SIGType

This command sets or queries the type of input signal used for measurement SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

**Conditions** This command is only valid if the instrument has Option MTM is installed.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:SOURCE<x>:SIGType {PULSE|EYE}  
MEASUREMENT:IMMED:SOURCE<x>:SIGType?

**Arguments** PULSE is for generic signals that are not associated with synchronous communications standards.

EYE is for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).

**Examples** MEASUREMENT:IMMED:SOURCE1:SIGTYPE PULSE selects Pulse as the signal type for source 1.

MEASUREMENT:IMMED:SOURCE<X>:SIGTYPE? might return  
:MEASUREMENT:IMMED:SOURCE1:SIGTYPE EYE indicating that the signal type for source 1 is EYE.

## MEASUREMENT:IMMED:TYPE

This command sets or queries the immediate measurement type.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:TYPE {AMPLITUDE|AREA|BURST|CAREA|CMEAN|CRMS|DELAY|DISTDUTY|EXTINCTDB|EXTINCTPCT|EXTINCTRATIO|EYEHEIGHT|EYEWIDTH|FALL|FREQUENCY|HIGH|HITS|LOW|MAXIMUM|MEAN|MEDIAN|MINIMUM|NCROSS|NDUTY|NOVERSHOOT|NWIDTH|PBASE|PCROSS|PCTCROSS|PDUTY|PEAKHITS|PERIOD|PHASE|PK2PK|PKPKJITTER|PKPKNOISE|POVERSHOOT|PTOP|PWIDTH|QFACTOR|RISE|RMS|RMSJITTER|RMSNOISE|SIGMA1|SIGMA2|SIGMA3|SIXSIGMAJIT|SNRATIO|STDDEV| UNDEFINED|WAVEFORMS}  
MEASUREMENT:IMMED:TYPE?

**Arguments** AMPLITUDE measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

$$\text{Amplitude} = \text{High} - \text{Low}$$

AREA measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

BURST measures the duration of a burst. The measurement is made over the entire waveform or gated region.

CAREA (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

CMEAN (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

CRMS (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

DELAY measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

DISTDUTY (duty cycle distortion) measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level. It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

EXTINCTDB measures the extinction ratio of an optical waveform (eye diagram). Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction dB} = 10 \times (\log 10 (\text{High} / \text{Low}))$$



EXTINCTPCT measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction \%} = 100.0 \times (\text{Low} / \text{High})$$

EXTINCRATIO measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode. Extinction ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1  $\mu$ W.

$$\text{Extinction Ratio} = (\text{High} / \text{Low})$$

EYEheight measures the vertical opening of an eye diagram in volts.

EYEWidth measures the width of an eye diagram in seconds.

FALL measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

FREQuency measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

HIGH measures the High reference (100% level, sometimes called Topline) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

HITS (histogram hits) measures the number of points in or on the histogram box.

LOW measures the Low reference (0% level, sometimes called Baseline) of a waveform.

MAXimum finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. When histogram is selected with the [MEASUREMENT:METHOD](#) command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the right-most bin in horizontal histograms.

MEAN amplitude measurement finds the arithmetic mean over the entire waveform or gated region. When histogram is selected with the [MEASUREMENT:METHOD](#) command, the mean measurement measures the average of all acquired points within or on the histogram.

MEDian (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

**MINIMUM** finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. When histogram is selected with the **MEASUREMENT:METHOd** command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the left-most nonzero bin in the horizontal histograms.

**NCROSS** (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**NDUTy** (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

**NOVERshoot** (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

$$\text{Negative Overshoot} = (\text{Low} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

**NWIDth** (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

**PBASE** measures the base value used in extinction ratio measurements.

**PCROSS** (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**PCTCROSS** measures the location of the eye crossing point expressed as a percentage of **EYEHeight**.

$$\text{Crossing percent} = 100 \times [(\text{eye-crossing-point} - \text{PBASe}) / (\text{PTOP} - \text{PBASe})]$$

**PDUTy** (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

**PEAKHi ts** measures the number of points in the largest bin of the histogram.

**PERIOD** is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

**PHASE** measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

**PK2Pk** (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. When histogram is

selected with the **MEASUREMENT:METHOD** command, the PK2Pk measurement measures the histogram peak to peak difference.

**PKPKJitter** measures the variance (minimum and maximum values) in the time locations of the cross point.

**PKPKNoise** measures the peak-to-peak noise on a waveform at the mid reference level.

#### **POvershoot**

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{High}) / \text{Amplitude} \times 100\%$$

**PTOT** measures the top value used in extinction ratio measurements.

**PWidth** (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

**QFACTOR** measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

**RISe** timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

**RMS** amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

**RMSJitter** measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

**RMSNoise** measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

**SIGMA1** (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

**SIGMA2** (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

**SIGMA3** (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

**SIXSigmajitter** (histogram measurement) is  $6 \times \text{RMSJitter}$ .

**SNRatio** measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

**STDdev** measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

UNDEFINED is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.

WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

**Examples** MEASUREMENT:IMMED:TYPEFREQUENCY defines the immediate measurement to be a frequency measurement.

MEASUREMENT:IMMED:TYPE? might return :MEASUREMENT:IMMED:TYPE RMS, indicating that the immediate measurement is the true Root Mean Square voltage.

## MEASUREMENT:IMMED:UNITS? (Query Only)

This query-only command returns the units of the immediate measurement.

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:UNITS?

**Examples** MEASUREMENT:IMMED:UNITS? might return :MEASUREMENT:IMMED:UNITS "s", indicating that units for the immediate measurement are in seconds.

## MEASUREMENT:IMMED:VALUE? (Query Only)

This query-only command returns the value of the measurement specified by the [MEASUREMENT:IMMED:TYPE](#) command. The measurement is immediately taken on the source(s) specified by a [MEASUREMENT:IMMED:SOURCE<x>](#) command.

---

**NOTE.** A change to CH<x>:SCALE will not necessarily have taken effect it followed by this command.

---

**Group** Measurement

**Syntax** MEASUREMENT:IMMED:VALUE?

**Related Commands** [MEASUREMENT:IMMED:TYPE](#), [MEASUREMENT:IMMED:SOURCE<x>](#), \*ESR?, ALLEv?

**Examples** MEASUREMENT:IMMED:VALUE? might return :MEASUREMENT:IMMED:VALUE 9.9000E+37. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEv? commands.

## MEASUREMENT:MEAS<x>? (Query Only)

This query-only command returns all measurement parameters for the displayed measurement specified by x, which can range from 1 through 8. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Measurements table; then choosing the Time tab, clicking the Delay button and viewing the Delay Edge and Search Direction settings.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>?

**Examples** MEASUREMENT:MEAS1? might return :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:MEAS<x>:COUNT? (Query Only)

This query-only command returns the number of values accumulated for this measurement since the last statistical reset. Some values may have been ignored because they generated an error. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:COUNT?

**Examples** MEASUREMENT:MEAS3:COUNT? might return :MEASUREMENT:MEAS3:COUNT 3.247000000E+03.

## MEASUREMENT:MEAS<x>:DELAY? (Query Only)

This query-only command returns the delay measurement parameters for the measurement specified by <x>, which ranges from 1 through 8. Measurement parameters are presented in the following order: Edge1, Edge2, and Direction. This query command is equivalent to selecting Time from the Measure menu, choosing Delay and viewing the Delay Edge and Search Direction settings.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:DELAY?
<b>Examples</b>	MEASUREMENT:MEAS3? might return :MEASUREMENT:MEAS3:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS.

## MEASUREMENT:MEAS<x>:DELAY:DIRECTION

This command sets or queries the starting point and direction that determines the delay “to” edge when taking a delay measurement. Use the [MEASUREMENT:MEAS<x>:SOURCE<x>](#) command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and then clicking the desired Search Direction setting. Measurements are specified by x, which ranges from 1 through 8.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:DELAY:DIRECTION {BACKWARDS FORWARDS} MEASUREMENT:MEAS<x>:DELAY:DIRECTION?
<b>Related Commands</b>	<a href="#">MEASUREMENT:MEAS&lt;x&gt;:SOURCE&lt;x&gt;</a>
<b>Arguments</b>	<p>BACKWARDS means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. Use the <a href="#">MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt;</a> command to specify the slope of the edge.</p> <p>FORWARDS means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. Use the <a href="#">MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt;</a> command to specify the slope of the edge.</p>

**Examples** MEASUREMENT:MEAS1:DELAY:DIRECTION BACKWARDS starts searching from the end of the waveform record.

MEASUREMENT:MEAS3:DELAY:DIRECTION? might return  
:MEASUREMENT:MEAS3:DELAY:DIRECTION BACKWARDS, indicating that the current search direction is backward.

## MEASUREMENT:MEAS<x>:DELAY:EDGE<x>

This command sets or queries the slope of the edge that is used for the delay “from” waveform (edge1) and the delay “to” waveform (edge2) when taking a delay measurement. Use the MEASUREMENT:MEAS<x>:SOURCE2 command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and then clicking the desired Delay Edge setting. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:DELAY:EDGE<x> {FALL|RISE}  
MEASUREMENT:MEAS<x>:DELAY:EDGE<x>?

**Related Commands** [MEASUREMENT:MEAS<x>:SOURCE<x>](#)

**Arguments** FALL specifies the falling edge of the waveform.  
RISE specifies the rising edge of the waveform.

**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>:MAXIMUM? (Query Only)

This query-only command returns the maximum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax**    MEASUREMENT:MEAS<x>:MAXimum?

**Examples**    MEASUREMENT:MEAS3:MAXIMUM? might return  
:MEASUREMENT:MEAS3:MAXIMUM 4.18E-9.

## MEASUREMENT:MEAS<x>:MEAN? (Query Only)

This query-only command returns the mean value accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group**    Measurement

**Syntax**    MEASUREMENT:MEAS<x>:MEAN?

**Examples**    MEASUREMENT:MEAS1:MEAN? might return :MEASUREMENT:MEAS1:MEAN  
514.71E-09.

## MEASUREMENT:MEAS<x>:METHOD

This command specifies or queries the method used to calculate the 0% and 100% reference level.

**Group**    Measurement

**Syntax**    MEASUREMENT:MEAS<x>:METHOD {HISTogram|MINMax|MEAN}  
MEASUREMENT:MEAS<x>:METHOD?

**Arguments**    HISTogram sets the high and low waveform levels statistically using a histogram algorithm.

MINMax sets the high and low waveform levels to MAX and MIN, respectively.

MEAN sets the high and low waveform levels to their mean.

**Examples**    MEASUREMENT:MEAS1:METHOD HISTOGRAM specifies that the high and low reference levels are set statistically.

MEASUREMENT:MEAS1:METHOD? might return :MEASUREMENT:MEAS1:METHOD  
MINMAX, indicating that the reference levels are set to MIN and MAX.



## MEASUREMENT:MEAS<x>:MINIMUM? (Query Only)

This query-only command returns the minimum value found for this measurement since the last statistical reset. Measurements are specified by *x*, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:MINIMUM?

**Examples** MEASUREMENT:MEAS1:MINIMUM? might return  
:MEASUREMENT:MEAS1:MINIMUM 1.75E-09.

## MEASUREMENT:MEAS<x>:NOISE

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined. Measurements are specified by *x*, which ranges from 1 through 8.

**Conditions** This command is only valid if the instrument has Option MTM is installed.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:NOISE {HIGH|LOW}  
MEASUREMENT:MEAS<x>:NOISE?

**Arguments** HIGH argument causes the measurement for noise to be taken at the high level of the waveform.

LOW argument causes the measurement for noise to be taken at the low level of the waveform.

**Examples** MEASUREMENT:MEAS1:NOISE HIGH causes the noise measurement to be made at the high level of the waveform for Measurement 1.

MEASUREMENT:MEAS2:NOISE? might return :MEASUREMENT:MEAS2:NOISE LOW, indicating that Measurement 2 will measure noise at the low level of the waveform.

## MEASUREMENT:MEAS<x>:REFLevel? (Query Only)

This query-only command returns the current reference level parameters. It returns them in the following order: ABSOLUTE and then PERCENT for individual user measurements. Measurements are specified by x, which ranges from 1 through 8.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:REFLevel?
<b>Examples</b>	MEASUREMENT:MEAS1:REFLEVEL? might return :MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00, indicating that the reference level is set to 1.71 V.

## MEASUREMENT:MEAS<x>:REFLevel:ABSolute:HIGH

This command sets or queries the high reference level, and is the upper reference level when [MEASUREMENT:MEAS<x>:REFLevel:METHOD](#) is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of rise and fall measurements*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:REFLevel:ABSolute:HIGH <NR3> MEASUREMENT:MEAS<x>:REFLevel:ABSolute:HIGH?
<b>Related Commands</b>	<a href="#">MEASUREMENT:MEAS&lt;x&gt;:REFLevel:METHOD</a>
<b>Arguments</b>	<NR3> is the high reference level, in volts. The default is 0.0 V.
<b>Examples</b>	MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH 1.71 sets the high reference level to 1.71 V.  MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH? might return :MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00, indicating that the absolute high reference level is set to 1.71 V.

## MEASUREMENT:MEAS<x>:REFLevel:ABSolute:LOW

This command sets or queries the low reference level, and is the lower reference level when [MEASUREMENT:MEAS<x>:REFLevel:METHOD](#) is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of rise and fall measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:REFLevel:ABSolute:LOW <NR3>  
MEASUREMENT:MEAS<x>:REFLevel:ABSolute:LOW?

**Related Commands** [MEASUREMENT:MEAS<x>:REFLevel:METHOD](#)

**Arguments** <NR3> is the low reference level, in volts. The default is 0.0 V.

**Examples** MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW 0.0 V sets the low reference level to 0.0 V.

MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW? might return  
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW 0.0000E+00, indicating that the absolute low reference level is set to 0.0 V.

## MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID<x>

This command sets or queries the mid reference level for the “to” waveform when taking a delay measurement, and is the 50% reference level when [MEASUREMENT:MEAS<x>:REFLevel:METHOD](#) is set to Absolute. Mid1 sets the “from” waveform and Mid2 sets the “to” waveform when taking a delay measurement. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of period, frequency, delay, and all cyclic measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID<x> <NR3>  
MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID<x>?

**Related Commands**    [MEASUREMENT:MEAS<x>:REFLEVEL:METHOD](#)

**Arguments**    <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Examples**    MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2? might return  
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2 0.5000E+00, indicating that the absolute mid2 reference level is set to 0.5 V.

## MEASUREMENT:MEAS<x>:REFLEVEL:METHOD

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired reference level from the Units group box. Measurements are specified by x, which ranges from 1 through 8.

**Group**    Measurement

**Syntax**    MEASUREMENT:MEAS<x>:REFLEVEL:METHOD {ABSOLUTE|PERCENT}  
MEASUREMENT:MEAS<x>:REFLEVEL:METHOD?

**Arguments**    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 (for example, when designing to published interface specifications, such as RS-232-C).

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.

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

MEASUREMENT:MEAS1:REFLEVEL:METHOD? might return  
:MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

## MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the high reference level when [MEASUREMENT:MEAS<x>:REFLEVEL:METHOD](#) is set to Percent. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of rise and fall measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH <NR3>  
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH?

**Related Commands** [MEASUREMENT:MEAS<x>:REFLEVEL:METHOD](#)

**Arguments** <NR3> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

**Examples** MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 95 sets the high reference level to 95% of HIGH.

MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH? might return  
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90, indicating that the percentage high reference level is set to 90% of HIGH.

## MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the low reference level when [MEASUREMENT:MEAS<x>:REFLEVEL:METHOD](#) is set to Percent. This command is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Low Ref value. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of rise and fall measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW <NR3>  
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW?

**Related Commands**    [MEASUREMENT:MEAS<x>:REFLEVEL:METHod](#)

**Arguments**    <NR3> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

**Examples**    MEASUREMENT:MEAS1:REFLEVEL:PERCENT:LOW 15 sets the high reference level to 15% of HIGH.

MEASUREMENT:MEAS1:REFLEVEL:PERCENT:LOW? might return  
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:LOW 10, indicating that the percentage high reference level is set to 10% of HIGH.

## MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID<x>

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level for the second waveform specified when [MEASUREMENT:MEAS<x>:REFLEVEL:METHod](#) is set to Percent. Mid1 specifies the “from” waveform and Mid2 specifies the “to” waveform for delay measurements. Measurements are specified by x, which ranges from 1 through 8.

---

**NOTE.** *This command affects the results of delay measurements.*

---

**Group**    Measurement

**Syntax**    MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID<x> <NR3>  
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID<x>?

**Related Commands**    [MEASUREMENT:MEAS<x>:REFLEVEL:METHod](#)

**Arguments**    <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Examples**    MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2 40 sets the mid2 reference level to 40% of HIGH.

MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2? might return  
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2 45, indicating that the percentage mid2 reference level is set to 45% of HIGH.

## MEASUREMENT:MEAS<x>:SOURCE<x>

This command sets or queries the source for all single channel measurements and specifies the reference source to measure “to” when taking a delay measurement or phase measurement. Measurements are specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu, selecting a measurement type of either Phase or Delay, and then choosing the desired measurement source.

Tip: Source2 measurements apply only to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:SOURCE<x>  
{CH<x> | MATH<y> | REF<x> | HISTOGRAM}  
MEASUREMENT:MEAS<x>:SOURCE<x>?

**Related Commands** [MEASUREMENT:MEAS<x>:TYPE](#)

**Arguments** CH<x> is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4.

MATH<y> is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4.

REF<x> is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4.

HISTOGRAM is a histogram. Histogram is valid only for source 1.

**Examples** MEASUREMENT:MEAS4:SOURCE2 CH1 specifies CH1 as the delay “to” source when making delay measurement.

MEASUREMENT:MEAS2:SOURCE2? might return  
:MEASUREMENT:MEAS2:SOURCE2 MATH1, indicating that Math 1 is the measurement 2 source.

## MEASUREMENT:MEAS<x>:SOURCE<x>:SIGTYPE

This command sets or queries the type of input signal used for the specified measurement slot. MEAS<x> is 1 through 8 for the measurement slot using SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed.
<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:SOURCE<x>:SIGTYPE {PULSE EYE} MEASUREMENT:MEAS<x>:SOURCE<x>:SIGTYPE?
<b>Arguments</b>	PULSE is for generic signals that are not associated with synchronous communications standards.  EYE is for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).
<b>Examples</b>	MEASUREMENT:MEAS5:SOURCE1:SIGTYPE PULSE selects Pulse as the signal type associated with source 1 for measurement slot 5.  MEASUREMENT:MEAS2:SOURCE1:SIGTYPE? might return :MEASUREMENT:MEAS2:SOURCE1:SIGTYPE EYE, indicating that the signal type associated with source 1 of measurement slot 2 is EYE.

## MEASUREMENT:MEAS<x>:STATE

This command sets or queries whether the specified measurement slot is computed and displayed. The measurement slot is specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then clicking the Display button.

For a measurement to display, you must have selected a source waveform and defined the measurement you want to take and display. You select the measurement using the [MEASUREMENT:MEAS<x>:SOURCE<x>](#) command. You define the measurement type using the [MEASUREMENT:MEAS<x>:TYPE](#) command.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:MEAS<x>:STATE {OFF ON <NR1>} MEASUREMENT:MEAS<x>:STATE?
<b>Related Commands</b>	<a href="#">MEASUREMENT:MEAS&lt;x&gt;:SOURCE&lt;x&gt;</a> , <a href="#">MEASUREMENT:MEAS&lt;x&gt;:TYPE</a>



- Arguments** OFF disables calculation and display of the specified measurement slot.  
 ON enables calculation and display of the specified measurement slot.  
 <NR1> = 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.
- Examples** MEASUREMENT:MEAS2:STATE ON computes and displays the measurement defined as measurement 2.  
 MEASUREMENT:MEAS1:STATE? might return :MEASUREMENT:MEAS1:STATE 0, indicating that measurement defined for measurement slot 1 is disabled.

## MEASUREMENT:MEAS<x>:STDdev? (Query Only)

This query-only command returns the standard deviation of values accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

- Group** Measurement
- Syntax** MEASUREMENT:MEAS<x>:STDdev?
- Examples** MEASUREMENT:MEAS1:STDDEV? might return :MEASUREMENT:MEAS1:STDDEV 21.0E-12.

## MEASUREMENT:MEAS<x>:TYPE

This command sets or queries the measurement type defined for the specified measurement slot. The measurement slot is specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement type.

- Group** Measurement
- Syntax** MEASUREMENT:MEAS<x>:TYPE {AMPLITUDE|AREA|BURST|CAREA|CMean|CRMS|DELAY|DISTDuty|EXTINCTDB|EXTINCTPCT|EXTINCTRATIO|EYEHEIGHT|EYEWIDTH|FALL|FREQUENCY|HIGH|HITS|LOW|MAXIMUM|MEAN|MEDIAN|MINIMUM|NCROSS|NDuty|NOVERSHOOT|NWIDTH|PBASE|PCROSS|PCTCROSS|PDuty|PEAKHITS|PERIOD|PHASE|PK2PK|PKPKJITTER|PKPKNOISE|POVERSHOOT|PTOP|PWIDTH|QFACTOR|

RISe|RMS|RMSJitter|RMSNoise|SIGMA1|SIGMA2|  
 SIGMA3|SIXSigmajit|SNRatio|STDdev|UNDEFINED| WAVEFORMS}  
 MEASUREMENT:MEAS<x>:TYPE?

## Arguments

**AMPLitude** measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

$$\textit{Amplitude} = \textit{High} - \textit{Low}$$

**AREa** measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

**BURSt** measures the duration of a burst. The measurement is made over the entire waveform or gated region.

**CAREa** (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

**CMEan** (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

**CRMS** (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

**DELAy** measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

**DISTDUty** (duty cycle distortion) measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level. It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

**EXTINCTDB** measures the extinction ratio of an optical waveform (eye diagram). Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\textit{Extinction dB} = 10 \times (\log 10 (\textit{High} / \textit{Low}))$$

**EXTINCTPCT** measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\textit{Extinction \%} = 100.0 \times (\textit{Low} / \textit{High})$$

**EXTINCRATIO** measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode. Extinction ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1  $\mu$ W.

$$\text{Extinction Ratio} = (\text{High} / \text{Low})$$

**EYEHeight** measures the vertical opening of an eye diagram in volts.

**EYEWidth** measures the width of an eye diagram in seconds.

**FALL** measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

**FREQUENCY** measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

**HIGH** measures the High reference (100% level, sometimes called Topline) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

**HITS** (histogram hits) measures the number of points in or on the histogram box.

**LOW** measures the Low reference (0% level, sometimes called Baseline) of a waveform.

**MAXimum** finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. When histogram is selected with the **MEASUREMENT:METHOD** command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the right-most bin in horizontal histograms.

**MEAN** amplitude measurement finds the arithmetic mean over the entire waveform or gated region. When histogram is selected with the **MEASUREMENT:METHOD** command, the mean measurement measures the average of all acquired points within or on the histogram.

**MEDian** (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

**MINimum** finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. When histogram is selected with the **MEASUREMENT:METHOD** command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the left-most nonzero bin in the horizontal histograms.

**NCROSS** (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**NDuty** (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

**NOvershoot** (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

$$\text{Negative Overshoot} = (\text{Low} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

**NWidth** (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

**PBASE** measures the base value used in extinction ratio measurements.

**PCROSS** (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**PCTCROSS** measures the location of the eye crossing point expressed as a percentage of **EYEHeight**.

$$\text{Crossing percent} = 100 \times [(\text{eye-crossing-point} - \text{PBASe}) / (\text{PTOP} - \text{PBASe})]$$

**PDuty** (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

**PEAKHits** measures the number of points in the largest bin of the histogram.

**PERIOD** is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

**PHASE** measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

**PK2Pk** (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. When histogram is selected with the **MEASUREMENT:METHOD** command, the PK2Pk measurement measures the histogram peak to peak difference.

**PKPKJitter** measures the variance (minimum and maximum values) in the time locations of the cross point.

**PKPKNoise** measures the peak-to-peak noise on a waveform at the mid reference level.

#### **POvershoot**

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{High}) / \text{Amplitude} \times 100\%$$

**PTOT** measures the top value used in extinction ratio measurements.

**PWidth** (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

**QFactor** measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

**RISe** timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

**RMS** amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

**RMSJitter** measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

**RMSNoise** measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

**SIGMA1** (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

**SIGMA2** (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

**SIGMA3** (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

**SIXSigmajit** (histogram measurement) is  $6 \times \text{RMSJitter}$ .

**SNRatio** measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

**STDdev** measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

**UNDEFINED** is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.

WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

- 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>:UNITS? (Query Only)

This query-only command returns the units associated with the specified measurement. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:UNITS?

**Related Commands** [MEASUREMENT:MEAS<x>:TYPE](#)

- Examples** MEASUREMENT:MEAS1:UNITS? might return :MEASUREMENT:MEAS1:UNITS %, indicating units for measurement 1 are set to percent.

## MEASUREMENT:MEAS<x>:VALUE? (Query Only)

This query-only command returns the value that has been calculated for the measurement specified by <x>, which ranges from 1 through 8. This command is equivalent to selecting Display Statistics from the Measure menu and then choosing Value from the drop-down list to display *all* measurement values on-screen.

---

**NOTE.** *This is the displayed value in the on-screen display. If measurement statistics are enabled, a new value is calculated with every waveform. In addition, this value is updated about every 1/3 second. If you are acquiring a long acquisition record, the instrument may take longer to update.*

---

**Group** Measurement

**Syntax** MEASUREMENT:MEAS<x>:VALUE?

**Related Commands** [MEASUREMENT:MEAS<x>:UNITS?](#), [\\*ESR?](#), [ALLEV?](#)

**Examples** `MEASUREMENT:MEAS1:VALUE?` might return `:MEASUREMENT:MEAS1:VALUE 2.8740E-06`. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the [\\*ESR?](#) and [ALLEV?](#) commands.

## MEASUREMENT:METHOd

This command sets or queries the method used to calculate the 0% and 100% reference level. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired Determine Base, Top From setting.

**Group** Measurement

**Syntax** `MEASUREMENT:METHOd {HISTogram|MEAN|MINMax}`  
`MEASUREMENT:METHOd?`

**Related Commands** [MEASUREMENT:REFLevel:PERCent:HIGH](#), [MEASUREMENT:REFLevel:PERCent:LOW](#), [MEASUREMENT:REFLevel:PERCent:MID<x>](#),

**Arguments** `HISTogram` sets the high and low reference levels to the most common values either above or below the mid point, depending on whether the high reference point or the low reference point is being defined. Because the statistical approach ignores short-term aberrations, such as overshoot or ringing, the histogram method is the best setting for examining pulses.

`MEAN` sets the high and low reference levels to the mean values using all values either above or below the midpoint, depending of whether it is defining the high or low reference level. The selection is best used for examining eye patterns.

`MINMax` uses the highest and lowest values of the waveform record. This selection is best for examining waveforms with no large, flat portions of a common value, such as sine waves and triangle waves.

**Examples** `MEASUREMENT:METHOdHISTOGRAM` specifies that the high and low reference levels are set statistically.

`MEASUREMENT:METHOd?` might return `:MEASUREMENT:METHOd MINMAX`, indicating that the reference levels are set to MIN and MAX.

## MEASUREMENT:NOISE

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined.

**Conditions** This command is only valid if the instrument has Option MTM is installed

**Group** Measurement

**Syntax** MEASUREMENT:NOISE {HIGH|LOW}  
MEASUREMENT:NOISE?

**Arguments** HIGH argument causes the measurement for noise to be taken at the high level of the waveform.  
  
LOW argument causes the measurement for noise to be taken at the low level of the waveform.

**Examples** MEASUREMENT:NOISEHIGH causes the noise measurement to be made at the high level of the waveform.  
  
MEASUREMENT:NOISE? might return :MEASUREMENT:NOISE LOW, indicating that the noise measurement will be made at the low level of the waveform.

## MEASUREMENT:REFLevel:ABSolute:HIGH

This command sets or queries the high reference level, and is the upper reference level when [MEASUREMENT:REFLevel:METHOD](#) is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute High Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASUREMENTS:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLevel commands.*

---

**Group** Measurement



**Syntax** MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH <NR3>  
MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH?

**Related Commands** MEASUREMENT:REFLEVEL:METHOD, MEASUREMENT:IMMED:TYPE, MEASUREMENT:MEAS<x>:TYPE

**Arguments** <NR3> is the high reference level, in volts. The default is 0.0 V.

**Examples** MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH1.71 sets the high reference level to 1.71 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH? might return  
:MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00,  
indicating that the absolute high reference level is set to 1.71 V.

## MEASUREMENT:REFLEVEL:ABSOLUTE:LOW

This command sets or queries the low reference level, and is the lower reference level when [MEASUREMENT:REFLEVEL:METHOD](#) is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Low Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.*

---

**Group** Measurement

**Syntax** MEASUREMENT:REFLEVEL:ABSOLUTE:LOW <NR3>  
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW?

**Related Commands** MEASUREMENT:REFLEVEL:METHOD, MEASUREMENT:IMMED:TYPE, MEASUREMENT:MEAS<x>:TYPE

**Arguments** <NR3> is the low reference level, in volts. The default is 0.0 V.

**Examples** MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0 sets the low reference level to 0.0 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:LOW? might return  
 :MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0000E+00, indicating that the  
 absolute low reference level is set to 0.0 V.

## MEASUREMENT:REFLEVEL:ABSOLUTE:MID<x>

This command sets or queries the mid reference level for the “from” waveform (mid1) and the “to” waveform (mid2) when taking a delay measurement, and is the 50% reference level when [MEASUREMENT:REFLEVEL:METHOD](#) is set to Absolute. This command affects the results of period, frequency, delay, and all cyclic measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Mid2 Ref value.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.*

---

**Group** Measurement

**Syntax** MEASUREMENT:REFLEVEL:ABSOLUTE:MID<x> <NR3>  
 MEASUREMENT:REFLEVEL:ABSOLUTE:MID<x>?

**Related Commands** [MEASUREMENT:REFLEVEL:METHOD](#)

**Arguments** <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Examples** MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID2? might return  
 :MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5000E+00, indicating that the  
 absolute mid2 reference level is set to 0.5 V.

## MEASUREMENT:REFLEVEL:METHOD

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu, and then choosing the desired reference level from the Units group box.

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:REFLevel:METHOD {ABSolute PERCent} MEASUrement:REFLevel:METHOD?
<b>Arguments</b>	<p>ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REFLevel:ABSolute commands. This method is useful when precise values are required (for example, when designing to published interface specifications, such as RS-232-C).</p> <p>PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REFLevel:PERCent commands.</p>
<b>Examples</b>	<p>MEASUREMENT:REFLEVEL:METHODABSOLUTE specifies that explicit user-defined values are used for the reference levels.</p> <p>MEASUREMENT:REFLEVEL:METHOD? might return :MEASUREMENT:REFLEVEL:METHOD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.</p>

## MEASUrement:REFLevel:PERCent:HIGH

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the high reference level when MEASUrement:REFLevel:METHOD is set to Percent. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage High Ref value.

**NOTE.** This command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

**Group** Measurement

**Syntax**      MEASUREMENT:REFLEVEL:PERCENT:HIGH <NR3>  
 MEASUREMENT:REFLEVEL:PERCENT:HIGH?

**Related Commands**      MEASUREMENT:REFLEVEL:METHOD, MEASUREMENT:IMMED:TYPE,  
 MEASUREMENT:MEAS<x>:TYPE

**Arguments**      <NR3> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

**Examples**      MEASUREMENT:REFLEVEL:PERCENT:HIGH95 sets the high reference level to 95% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:HIGH? might return  
 :MEASUREMENT:REFLEVEL:PERCENT:HIGH 90, indicating that the percentage high reference level is set to 90% of HIGH.

## MEASUREMENT:REFLEVEL:PERCENT:LOW

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the low reference level when [MEASUREMENT:REFLEVEL:METHOD](#) is set to Percent. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Low Ref value.

---

**NOTE.** This command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.

---

**Group**      Measurement

**Syntax**      MEASUREMENT:REFLEVEL:PERCENT:LOW <NR3>  
 MEASUREMENT:REFLEVEL:PERCENT:LOW?

**Related Commands**      MEASUREMENT:REFLEVEL:METHOD, MEASUREMENT:IMMED:TYPE,  
 MEASUREMENT:MEAS<x>:TYPE

**Arguments**      <NR3> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

**Examples** MEASUREMENT:REFLEVEL:PERCENT:LOW15 sets the high reference level to 15% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:LOW? might return  
:MEASUREMENT:REFLEVEL:PERCENT:LOW 10, indicating that the percentage high reference level is set to 10% of HIGH.

## MEASUREMENT:REFLEVEL:PERCENT:MID<x>

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level when [MEASUREMENT:REFLEVEL:METHOD](#) is set to Percent. Mid1 specifies the percent for the first waveform and Mid2 specifies the percent for the second waveform specified for delay measurements. This command affects the results of period, frequency, delay, and all cyclic measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Mid Ref value.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.*

---

**Group** Measurement

**Syntax** MEASUREMENT:REFLEVEL:PERCENT:MID<x> <NR3>  
MEASUREMENT:REFLEVEL:PERCENT:MID<x>?

**Related Commands** [MEASUREMENT:REFLEVEL:METHOD](#)

**Arguments** <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Examples** MEASUREMENT:REFLEVEL:PERCENT:MID2 40 sets the mid2 reference level to 40% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:MID2? might return  
:MEASUREMENT:REFLEVEL:PERCENT:MID2 45, indicating that the percentage mid2 reference level is set to 45% of HIGH.

## MEASUREMENT:SOURCE<x>:SIGType

This command sets or queries the type of input signal used for measurement SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

**Conditions** This command is valid only if the instrument has Option MTM is installed

**Group** Measurement

**Syntax** MEASUREMENT:SOURCE<x>:SIGType {PULSE|EYE}  
MEASUREMENT:SOURCE<x>:SIGType?

**Arguments** PULSE is for generic signals that are not associated with synchronous communications standards.  
  
EYE is for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).

**Examples** MEASUREMENT:SOURCE1:SIGTYPE PULSE selects Pulse as the signal type for source 1.  
  
MEASUREMENT:SOURCE1:SIGTYPE? might return  
:MEASUREMENT:SOURCE1:SIGTYPE EYE, indicating that the signal type for source 1 is EYE.

## MEASUREMENT:STATISTICS:COUNT (No Query Form)

This command (no query form) clears existing measurement statistics from memory. This command is equivalent to selecting Measurement Setup from the Measure menu, selecting Statistics, and clicking the Reset button.

**Group** Measurement

**Syntax** MEASUREMENT:STATISTICS:COUNT {RESET}

**Related Commands** [MEASUREMENT:STATISTICS:MODE](#)

**Arguments** RESET clears existing measurement statistics from memory.

**Examples** MEASUREMENT:STATISTICS:COUNT RESET resets the count of the statistics measurement to zero, clears the measurement statistics from memory, and restarts the calculation.

## MEASUREMENT:STATISTICS:MODE

This command controls the operation and display of measurement statistics. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and then choosing the desired Measurement Format.

**Group** Measurement

**Syntax** MEASUREMENT:STATISTICS:MODE  
{OFF|ALL|VALUEMean|MINMax|MEANSTDdev}  
MEASUREMENT:STATISTICS:MODE?

**Related Commands** [MEASUREMENT:STATISTICS:WEIGHTING](#)

**Arguments** OFF turns off all measurements. This is the default value.

ALL turns on statistics and displays all statistics for each measurement.

VALUEMean turns on statistics and displays the value and the mean ( $\mu$ ) of each measurement.

MINMax turns on statistics and displays the min and max of each measurement.

MEANSTDdev turns on statistics and displays the mean and standard deviation of each measurement.

**Examples** MEASUREMENT:STATISTICS:MODEOFF turns off statistic measurements.

MEASUREMENT:STATISTICS:MODE? might return  
:MEASUREMENT:STATISTICS:MODE ALL, indicating that measurement statistics are turned on and all statistics are being displayed for each measurement.

## MEASUREMENT:STATISTICS:WEIGHTING

This command sets or queries the time constant for mean and standard deviation statistical accumulations. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and entering the desired Weight n= value.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:STATISTICS:WEIGHTING <NR1> MEASUREMENT:STATISTICS:WEIGHTING?
<b>Related Commands</b>	<a href="#">MEASUREMENT:STATISTICS:MODE</a>
<b>Arguments</b>	<NR1> is the time constant for the mean and standard deviation statistical accumulations.
<b>Examples</b>	MEASUREMENT:STATISTICS:WEIGHTING4 sets statistical weighting to 4.  MEASUREMENT:STATISTICS:WEIGHTING? might return :MEASUREMENT:STATISTICS:WEIGHTING 4, indicating that measurement statistics weighting is currently set to 4.

## 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.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	NEWpass <QString>
<b>Related Commands</b>	<a href="#">PASSWORD</a> , <a href="#">*PUD</a>
<b>Arguments</b>	<QString> is the new password, which can contain up to 10 characters.
<b>Examples</b>	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.

**Table 2-45: Commands that Generate an OPC Message**

Operation	Command
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN or ACQuire:STATE 1 (when ACQuire:STOPAfter is set to SEQuence)
Starts diagnostics	DIAG:EXECUTE
Calibration step	Refer to the Service Manual.
Test execution	TEST
Recall setup	*RST, FACtory, TEKSecure *RCL, RECAL:SETUp

**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 list of the options installed in your instrument.

**Group** Status and Error

**Syntax** \*OPT?

**Examples** \*OPT? might return :3M:16 Meg Max 4M/CH.

## 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.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	PASSWord <QString>
<b>Related Commands</b>	<a href="#">NEWpass</a> , <a href="#">*PUD</a>
<b>Arguments</b>	<QString> is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.
<b>Examples</b>	<p>PASSWORD "XYZZY" enables the *PUD and NEWPass set commands.</p> <p>PASSWORD disables the *PUD and NEWPass set commands. You can still use the query version of *PUD.</p>

## \*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.

<b>Group</b>	Status and Error
<b>Syntax</b>	<pre>*PSC {OFF ON &lt;NR1&gt;} *PSC?</pre>
<b>Related Commands</b>	<a href="#">DESE</a> , <a href="#">*ESE</a> , <a href="#">FACtory</a> , <a href="#">*RST</a> , <a href="#">*SRE</a>

**Arguments** 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.

<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.

**Examples** \*PSC0 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.

## \*RCL (No Query Form)

This command (no query form) restores the state of the instrument from a copy of the settings stored in memory (The settings are stored using the \*SAV command). If ‘factory’ is referenced (by specifying ‘0’), the factory default values will be restored.

This command is equivalent to RECALL:SETup and performs the same function as selecting Recall from the File menu, and then pressing the Setup button.

**Group** Save and Recall

**Syntax** \*RCL <NR1>

**Related Commands** [DELEte:SETUp](#), [FACtory](#), [\\*LRN?](#), [RECALL:SETUp](#), [\\*RST](#), [\\*SAV](#), [SAVE:SETUp](#)

**Arguments** <NR1> specifies a setup storage location value ranging from 0 through 10. Using an out-of-range value causes an execution error.

**Examples** \*RCL 3 restores the instrument from a copy of the settings stored in memory location 3.

## RECALL:MASK (No Query Form)

This command (no query form) recalls a stored mask as the User mask.

**Group** Mask

**Syntax** RECALL:MASK <QString>

**Related Commands** [SAVE:MASK](#)

**Arguments** <QString> is a quoted string that defines the file name and path from where the mask will be recalled. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the mask from the default directory. <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.

The file is stored in ASCII format and is user-editable.

**Examples** RECALL:MASK "TEST" recalls the mask stored in the file named TEST as the User mask.

## 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. If factory is referenced (by specifying '0'), the factory default values will be restored. This command is equivalent to \*RCL and 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|<NR1>|<file path>}

**Related Commands** [FACTory](#), [\\*RCL](#), [\\*RST](#), [\\*SAV](#), [SAVE:SETUp](#)

**Arguments** FACTory restores the factory setup.

<NR1> specifies a setup storage location to restore. Setup storage location values range from 0 through 10; using an out of range value causes an error (222, "Data out of range").

<file path> specifies a location for an instrument setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the file from the default directory (C:\TekScope\setups). <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required, but highly recommended.

**Examples** RECALL:SETUPFACTORY recalls (and makes current) the instrument setup to its factory defaults.

RECALL:SETUP2 recalls the instrument setup from setup storage location 2.

RECALL:SETUP "TEK00000.SET" recalls the setup from the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).

## RECALL:SETUp:DESKew

This command sets or queries the deskew values that are affected by a default setup or a recalled setup.

---

**NOTE.** *This setting is changed only by TekSecure (which resets it to the default setting of UNLOCK) or through explicit user request (from the Recall Setup dialog box in the TekScope User Interface).*

---

<b>Group</b>	SaveRecall
<b>Syntax</b>	RECALL:SETUP:DESKEW {LOCK UNLOCK} RECALL:SETUP:DESKEW?
<b>Arguments</b>	<p>LOCK a default or a recall setup will not change the deskew settings when this option is selected.</p> <p>UNLOCK a default setup will reset the deskew settings to factory values and a recall setup will apply the setup deskew values to the recalled oscilloscope state.</p>
<b>Examples</b>	<p>RECALL:SETUP:DESKEW LOCK, a default or recall setup will not change the deskew settings.</p> <p>RECALL:SETUP:DESKEW? might return :RECALL:SETUP:DESKEW UNLOCK, indicating that a default setup will reset the deskew settings to factory values and a recall setup will apply the setup deskew values to the recalled oscilloscope state.</p>

## RECALL:WAVEform (No Query Form)

This command (no query form) recalls a stored waveform to a reference location. This command is equivalent to selecting Recall from the File menu, and then pressing the Waveform button.

<b>Group</b>	Save and Recall
<b>Syntax</b>	RECALL:WAVEform <file path>,REF<x>
<b>Related Commands</b>	<a href="#">DELETE:WAVEform</a> , <a href="#">SAVE:WAVEform</a>
<b>Arguments</b>	<p>&lt;file path&gt; specifies a location for a stored waveform file. &lt;file path&gt; is a quoted string that defines the file name and path. Input the file path using the form &lt;drive&gt;:&lt;dir&gt;/&lt;filename&gt;. &lt;drive&gt; and one or more &lt;dir&gt;s are optional. If you do not specify them, the instrument will read the waveform from the default directory (C:\TekScope\Waveforms). &lt;filename&gt; stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.</p> <p>REF&lt;x&gt; specifies a location in internal reference memory. Reference memory location values range from 1 through 4.</p>

**Examples** `RECALL:WAVEFORM "TEK00000.WFM"`, REF1 recalls the waveform stored in the file named `TEK00000.WFM` from the default directory for waveforms (`C:\TekScope\Waveforms`) to the reference location 1.

## REF<x>:HORizontal:POsition

This command sets or queries the horizontal display position of the reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup¼ from the drop-down list, selecting a reference waveform, and then entering the horizontal position value using the multipurpose knob.

**Group** Vertical

**Syntax** `REF<x>:HORizontal:POsition <NR3>`  
`REF<x>:HORizontal:POsition?`

**Arguments** `<NR3>` argument specifies the horizontal position of the specified reference waveform in percentage of the waveform that is displayed to the right of the center vertical graticule. The range of this argument is from 0 through 100.

**Examples** `REF2:HORIZONTAL:POSITION 0.0` positions the start of the Reference 2 waveform at the center graticule.

`REF1:HORIZONTAL:POSITION?` might return `:REF1:HORIZONTAL:POSITION 50`, indicating that the Reference 1 waveform is centered on the horizontal graticule.

## REF<x>:LABel? (Query Only)

This query-only command returns a branch query containing the waveform label name and the coordinates at which the label (attached to the displayed waveform of the specified reference) is displayed. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Label from the drop-down list.

**Group** Vertical

**Syntax** `REF<x>:LABel?`

**Related Commands**    [REF<x>:LABel:NAME](#), [REF<x>:LABel:XPOS](#), [REF<x>:LABel:YPOS](#)

**Examples**    REF1:LABEL? might return :REF1:LABEl:NAME  
"Myname"; :XPOS-200; :YPOS 50.

## REF<x>:LABel:NAME

This command sets or queries the label of the designated waveform. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to create a label, and entering a label in the Label box.

**Group**    Vertical

**Syntax**    REF<x>:LABel:NAME <QString>  
REF<x>:LABel:NAME?

**Related Commands**    [REF<x>:LABel?](#)

**Arguments**    <QString> is the character string that will be used for the reference waveform label name.

**Examples**    REF4:LABEL:NAME "My Reference" sets the label name of Reference 4 waveform to "My Reference".

REF3:LABEL:NAME? might return :REF3:LABEl:NAME "Signal2", indicating that the label name for Reference 3 waveform is currently set to "Signal2".

## REF<x>:LABel:XPOS

This command sets or queries the X screen offset at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the screen. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to position a label, and entering a value in the X Position box.

**Group**    Vertical



**Syntax** REF<x>:LABel:XPOS <NR1>  
REF<x>:LABel:XPOS?

**Related Commands** [REF<x>:LABel?](#)

**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. Arguments should be integers ranging from 0 through 10.

**Examples** 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.

REF2:LABEL:XPOS? might return :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<x>:LABel:YPOS

This command sets or queries the Y screen offset at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the center of the screen. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to position a label, and entering a value in the Y Position box.

**Group** Vertical

**Syntax** REF<x>:LABel:YPOS <NR1>  
REF<x>:LABel:YPOS?

**Related Commands** [REF<x>:LABel?](#)

**Arguments** <NR1> is the location (in divisions) where the waveform label for the selected reference is displayed, relative to the waveform handle. Arguments should be integers ranging from 10 to -10.

**Examples** REF3:LABEL:YPOS -10 moves the waveform label for the Reference 3 waveform 10 divisions below the waveform handle.

REF2:LABEL:YPOS? might return :REF2:LABEL:YPOS 0, indicating that the waveform label for the Reference 2 waveform is currently located at the waveform handle.

## REF<x>:THRESHold

This command sets or queries the comparable threshold for converting the reference signal to digital form for the channel specified by x. The value of x can range from 1 through 4.

**Group** Digital

**Syntax** REF<x>:THRESHold <NR3>  
REF<x>:THRESHold?

**Arguments** <NR3> specifies the reference threshold, in volts.

**Examples** REF1:THRESHOLD 1 sets the threshold value of the reference REF1 to 1 volt.

REF1:THRESHOLD? might return REF1:THRESHOLD 500.0000E-3, indicating that the threshold value set for the reference REF1 is 0.5 volts.

## REF<x>:VERTical:POSition

This command sets or queries the vertical position of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4.

This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup from the drop-down list, selecting a reference waveform, and then entering the Position value using the multipurpose knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform. The position value determines the vertical graticule coordinate at which signal values are displayed. For example, if the position for Reference 3 is set to 2.0, the signal represented by that reference will be displayed at 2.0 divisions above the center of the screen.

---

**NOTE.** References are static. All position and scale changes are applied in a post-processing mode.

---

**Group** Vertical

**Syntax** REF<x>:VERTical:POsition <NR3>  
REF<x>:VERTical:POsition?

**Related Commands** CH<x>:POsition, MATH<x>:VERTical:POsition, D<x>:POsition

**Arguments** <NR3> is the desired position, control in divisions from the center horizontal graticule. The range is from -5 to 5 divisions.

**Examples** REF2:VERTICAL:POSITION 1.3E+00 positions the Reference 2 waveform input signal 1.3 divisions above the center horizontal graticule.

REF1:VERTICAL:POSITION? might return :REF1:VERTICAL:POSITION -1.3000E+00, indicating that the current position of Reference 1 waveform is 1.3 divisions below the center horizontal graticule.

## REF<x>:VERTical:SCALE

This command sets or queries the vertical scale of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup from the drop-down list, selecting a reference waveform, and then entering the Scale value using the multipurpose knob.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects reference and math waveforms differently from channel waveforms:

- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.
- For channel waveforms, this setting controls the vertical size of the acquisition window and the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

**Group** Vertical

**Syntax** REF<x>:VERTical:SCALE <NR3>  
REF<x>:VERTical:SCALE?

**Related Commands** [CH<x>:SCAlE](#), [MATH<x>:VERTical:SCAlE](#)

**Arguments** <NR3> is the gain in user units per division.

**Examples** REF4:VERTICAL:SCALE 100E-03 sets the Reference 4 waveform scale to 100 mV per division.

REF4:VERTICAL:SCALE? might return :REF2:VERTICAL:SCALE 1.0000e+00, indicating that the current vertical scale setting for Reference 2 waveform is 1 V per division.

## REM (No Query Form)

This command (no query form) embeds a comment within GPIB programs as a means of internally documenting the programs. The instrument ignores these embedded comment lines.

**Group** Miscellaneous

**Syntax** REM <QString>

**Related Commands** [NEWpass](#), [\\*PUD](#)

**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:OUT:FREQuency

This command sets or returns the selected frequency for the timebase reference output signal.

**Group** Miscellaneous

**Syntax** ROSc:OUT:FREQuency {MHZ10|MHZ100}  
ROSc:OUT:FREQuency?

**Arguments** MHZ10 outputs a 10 MHz reference signal to the REF OUT connector.  
 MHZ100 outputs a 100 MHz reference signal to the REF OUT connector.

**Examples** `ROSC:OUT:FREQUENCYMHZ10` outputs a 10 MHz reference signal to the REF OUT connector.  
`ROSC:OUT:FREQUENCY?` might return `ROSC:OUT:FREQUENCY MHZ100` indicating that the 100 MHz reference signal is output to the REF OUT connector.

## 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.

This command is also useful for synchronizing multiple instruments.

**Group** Miscellaneous

**Syntax** `ROSc:SOUrce {INTERNAL|EXTERNAL}`  
`ROSc:SOUrce {INTERNAL|EXTERNAL|TEKLink}`  
`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.  
`TEKLink` specifies the TekLink signal as the time base reference.

**Examples** `ROSC:SOURCEINTERNAL` 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`.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	ROSC:STATE?
<b>Related Commands</b>	<a href="#">ROSc:TRACking</a>
<b>Examples</b>	ROSC:STATE? might return :ROSC:STATE LOCKED, indicating that the time base reference is locked.

## ROSc:TRACking

DSA70000, DPO70000, DSA70000B, DPO70000B, and MSO70000 Series only:  
 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 signal that is fed through or bypasses the phase-locked loop.

This command is also useful for synchronizing multiple instruments.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	ROSc:TRACking {STABle FAST} ROSC:TRACking?
<b>Arguments</b>	STABle tracking mode specifies that the external reference signal is fed through a phase-locked loop that removes jitter from the external reference.  FAST tracking mode specifies that the external reference signal bypasses the phase-locked loop.
<b>Examples</b>	ROSC:TRACKINGSTABLE feeds the external reference signal through the instrument phase-locked loop.  ROSC:TRACKING? might return :ROSC:TRACKING FAST, indicating that the external reference signal bypasses the phase-locked loop.

## \*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings. The \*RST command does not change the following:

- The state of the IEEE Std 488.1-1987 interface
- The selected IEEE Std 488.1-1987 address of the instrument
- Calibration data that affect device specifications
- The Output Queue
- The Service Request Enable Register setting
- The Power-on status clear flag setting
- Alias definitions
- Stored settings
- The \*PUD? Response
- Any of the values associated with the DATA command.

This command is equivalent to pressing the **DEFAULT SETUP** button on the front panel.

**Group** Status and Error

**Syntax** \*RST

**Related Commands** [FACTory](#), [RECALL:SETUp](#), [SAVe:SETUp](#)

**Arguments** None

**Examples** \*RST resets the instrument settings to factory defaults.

## \*SAV (No Query Form)

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state.

This command is equivalent to selecting Save As from the File menu, and then pressing the Setup button.

<b>Group</b>	Save and Recall
<b>Syntax</b>	*SAV <NR1>
<b>Related Commands</b>	<a href="#">*RCL</a> , <a href="#">RECALL:SETUp</a> , <a href="#">SAVE:SETUp</a>
<b>Arguments</b>	<NR1> specifies a location in which to save the state of the instrument. Location values range from 1 through 10. Using an out-of-range location value causes an execution error. Any settings that have been stored previously at this location will be overwritten.
<b>Examples</b>	*SAV 2 saves the current instrument state in memory location 2.

## SAVE:MASK (No Query Form)

This command (no query form) saves the user mask to a mass storage file.

<b>Group</b>	Mask
<b>Syntax</b>	SAVE:MASK <QString>
<b>Related Commands</b>	<a href="#">RECALL:MASK</a>
<b>Arguments</b>	<p>&lt;QString&gt; is the location to where the mask is stored. &lt;file path&gt; is a quoted string that defines the file name and path. Input the file path using the form &lt;drive&gt;/&lt;dir&gt;/&lt;filename&gt;. &lt;drive&gt; and one or more &lt;dir&gt;s are optional.</p> <p>The file is stored in ASCII format and is user-editable.</p>
<b>Examples</b>	SAVE:MASK "TEST" saves the user mask to the file TEST in the default directory and on the default drive.

## SAVE:SETUp (No Query Form)

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state. This is equivalent to selecting Save As from the File menu, and then pressing the Setup button.



<b>Group</b>	Save and Recall
<b>Syntax</b>	SAVE:SETUp {<file path> <NR1>}
<b>Related Commands</b>	<a href="#">*RCL</a> , <a href="#">RECALL:SETUp</a> , <a href="#">*SAV</a>
<b>Arguments</b>	<p>&lt;file path&gt; is the target location for storing the setup file. &lt;file path&gt; is a quoted string that defines the file name and path. Input the file path using the form &lt;drive&gt;:&lt;dir&gt;/&lt;filename&gt;. &lt;drive&gt; and one or more &lt;dir&gt;s are optional. If you do not specify them, the instrument will store the file in the default directory (C:\TekScope\setups). &lt;filename&gt; stands for a filename of up to 128 characters. (Use of wildcard characters in filenames is not supported.) Filename extensions are not required but are highly recommended. For setups, use the extension “.SET”.</p> <p>&lt;NR1&gt; specifies a location for saving the current front panel setup. The front panel setup value ranges from 1 to 10. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.</p>
<b>Examples</b>	<p>SAVE:SETUP5 saves the current instrument setup in memory location 5.</p> <p>SAVE:SETUP"TEK00000.SET" saves the current instrument setup in the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).</p>

## SAVE:WAVEform (No Query Form)

This command (no query form) saves a waveform to one of four reference memory locations or a file. This command is equivalent to selecting Reference Waveforms from the File menu and choosing Save Wfm from the drop-down list.

<b>Group</b>	Save and Recall
<b>Syntax</b>	SAVE:WAVEform [<wfm>, REF<x>]   [<wfm>, <QString>]   [ALL, [QString]]   [DIGITALALL, <QString>]
<b>Related Commands</b>	<a href="#">DELEte:WAVEform</a> , <a href="#">RECALL:WAVEform</a>
<b>Arguments</b>	<wfm>, REF<x> saves the specified waveform to the specified reference memory location. <wfm> can be any of CH<x>, MATH<x>, and REF<x>.

<wfm>, <Qstring> saves the specified waveform to the file specified. The file is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will write the waveform to the default directory. <filename> stands for a filename of up to 125 characters, followed by a period (“.”) and the three-character extension. Internal format waveforms use the .wfm extension, spreadsheet format files use the .csv extension and MathCad format files use the .dat extension.

ALL, [Qstring] saves all displayed waveforms, excluding digital waveforms, to individual files based on the SAVE:WAVEFORM:FILEFORMAT setting. If a file name argument is specified, it is used as a prefix for each file followed by the waveform ID (such as CH1, MATH1, REF1). If no file name is specified, the default prefix "TDSWfm\_" is used, and the file is saved to the default directory.

DIGITALALL, <Qstring> saves all digital waveforms to a single file specified by the quoted string argument and based on the SAVE:WAVEFORM:FILEFORMAT setting. Possible output formats include SPREADSHEET.CSV, SPREADSHEET.TXT, and INTERNAL. Any other formats default to SPREADSHEET.CSV output.

**Examples** SAVE:WAVEFORMMATH2, REF2 saves the Math2 waveform in reference memory location2.

SAVE:WAVEFORMMATH1, "TEK0000.WFM" saves the Math1 waveform to the file TEK0000.WFM in the default directory and on the default drive.

## SAVE:WAVEform:FILEFormat

This command specifies or returns the file format for saved waveforms. Waveform header and timing information is included in the resulting file of non-internal formats. The instrument saves DPO waveforms as a 1000 x 502 matrix, with the first row corresponding to the most recently acquired data. The values specified by [DATA:START](#) and [DATA:STOP](#) determine the range of waveform data to output. In the event that DATA:STOP value is greater than the current record length, the current record length determines the last output value.

This command is equivalent to selecting Save As from the File menu, clicking the Waveform button, and choosing the desired waveform file format from the Save as type drop-down list.

**Group** Save and Recall

**Syntax** SAVE:WAVEform:FILEFormat  
 {INTERNa] | MATHCad | MATLab | SPREADSHEETCsvg |  
 SPREADSHEETTvt | TIMEStamp}  
 SAVE:WAVEform:FILEFormat?

**Related Commands**

[CURVe](#), [DATA](#), [DATA:START](#), [DATA:STOP](#), [SAVE:WAVEform](#),  
[WFMinpre:NR\\_Pt](#), [WFMOuppre:NR\\_Pt?](#)

**Arguments**

**INTERNAL** specifies that waveforms are saved in an internal format, using a .wfm filename extension. These files can be recalled as reference waveforms. When this argument is specified, the settings specified via the [DATA:START](#) and [DATA:STOP](#) commands have no meaning as the entire waveform is saved.

**MATHCad** specifies that waveforms are saved in MathCad format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MathCad format enables easy import of waveform data into MathCad or MATLAB. For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MathCad format header files use a \_hdr.dat extension.

**MATLab** specifies that waveforms are saved in Matlab format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MATLAB format enables easy import of waveform data into MathCad or MATLAB. For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MATLAB format header files use a \_hdr.dat extension.

**SPREADSHEETCSV** specifies that waveform data is saved in a format that contains comma delimited values. These waveform data files are named using the .csv filename extension. Saving waveforms in CSV format enables spreadsheet programs to import the data.

**SPREADSHEETtxt** specifies that waveform data is saved in a format that contains tab delimited values. These waveform data files are named using the .txt filename extension. Saving waveforms in this format enables spreadsheet programs to import the data.

**TIMESamp** specifies that timestamp data is saved in a format that contains comma delimited values. These data files are named using the .txt filename extension. Saving waveforms in this format enables spreadsheet programs to import the data.

**Examples**

**SAVE:WAVEFORM:FILEFORMATINTERNAL** specifies that the internal file format is the format used for saving waveforms.

**SAVE:WAVEFORM:FILEFORMAT?** might return **:SAVE:WAVEFORM:FILEFORMATINTERNAL**, indicating that waveforms are saved using the internal format.

**\*SDS (No Query Form)**

This command (no query form) changes the specified setup to reference the factory setup instead of the specific user setup slot. The content of the setup slot is unchanged, but the data will no longer be accessible to you.

This command is equivalent to selecting Delete from the File menu, and then choosing the specified setup.

**Group** Save and Recall

**Syntax** \*SDS <NR1>

**Related Commands** [DELEte:SETUp](#)

**Arguments** <NR1> specifies a user setup location to delete. Setup storage location values range from 1 through 10; using an out-of-range value causes an error.

**Examples** \*SDS2 changes setup slot 2 to reference the factory setup.

## SEARCH:SEARCH<x>

This command converts the specified search marks to the user marks. <x> can be 1 to 8.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x> <QString>  
SEARCH:SEARCH<x>?

**Arguments** <QString> specifies the user search marks.

**Examples** SEARCH:SEARCH1 TOUSER converts the Search 1 mark to the user marks.

SEARCH:SEARCH1? might return :SEARCH:SEARCH1 TOUSER, indicating that all the search1 marks are converted to user marks.

## SEARCH:SEARCH<x>:COpy (No Query Form)

This command (no query form) copies the search criteria to the trigger, the trigger criteria to a search, or copy search to search. <x> is the search number.

**Group** Search and Mark

<b>Syntax</b>	SEARCH:SEARCH<x>:COPY {SEARCHtotrigger TRIGgertosearch FROMSearch<x>}
<b>Arguments</b>	SEARCHtotrigger copies the search criteria to the trigger. TRIGgertosearch copies the trigger criteria to the search. FROMSearch<x> copies the search criteria to the search, where <x> is 1 to 8.
<b>Examples</b>	SEARCH:SEARCH1:COPY TRIGGERTOSEARCH copies the trigger criteria to the search 1 criteria.

## SEARCH:SEARCH<x>:STATE

This command sets the search state to on or off. <x> is the search number, which can be 1 to 8. The query form returns the search state.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:STATE {<NR1> OFF ON} SEARCH:SEARCH<x>:STATE?
<b>Arguments</b>	OFF or <NR1> = 0 sets the search state to off. ON or <NR1> ≠ 0 sets the search state to on.
<b>Examples</b>	SEARCH:SEARCH2:STATE ON sets the search state for search 2 on. SEARCH:SEARCH1:STATE? might return :SEARCH:SEARCH1:STATE 0, indicating that the search 1 is off.

## SEARCH:SEARCH<x>:TOTAL? (Query Only)

This query-only command returns the total number of matches for the search. The total number of matches may be more than the number of marks placed. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	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:DDRMemory:AUTOBitrate

This command sets or queries the auto bit rate value for a DDR Memory analysis search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOBitrate  
{AUTO|CUSTOM}  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOBitrate?

**Arguments** AUTO sets the auto bit rate to auto. When auto is selected, the instrument determines the bit rate from the applied signal.

CUSTOM sets the auto bit rate to custom. When custom is selected, you set the bit rate value.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOBITRATE CUSTOM sets the auto bit rate to custom and you can set the desired bit rate value.  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOBITRATE? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOBITRATE AUTO, indicating that the auto bit rate is set to auto.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOReflevel

This command sets or queries the reference levels for a DDR Memory analysis search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOReflevel  
{AUTO|CUSTOM}  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOReflevel?

**Arguments** AUTO sets the reference levels to auto. When auto is selected, the instrument determines the reference level from the applied signal.

CUSTOM sets the reference levels to custom. When custom is selected, you set the reference level

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOREFLEVEL CUSTOM sets the DDR reference levels to custom and you can set the desired reference level.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOREFLEVEL? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:AUTOREFLEVEL AUTO, indicating that the reference levels are set to auto.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:BITRate

This command sets or queries the data rate value for a DDR Memory analysis search to determine where to place a mark. You can set the bit rate only when AUTOBITRATE is Custom. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:BITRate <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:BITRate?

**Related Commands** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOBitRate

**Arguments** <NR3> specifies the DDR data rate in bits per second.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:BITRATE 8.0 +36 sets the DDR data rate to 8.0E+30 Mb/s.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:BITRATE? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:BITRATE 9.9000E+36, indicating that the DDR data rate is set to 9.9 E+30 Mb/s.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSActive

This command sets or queries whether the expected chip-select will be an active HIGH or LOW.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSActive {HIGH|LOW}  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSActive?

**Arguments** HIGH sets the search to look for an active HIGH.  
LOW sets the search to look for an active HIGH.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSACTIVE HIGH sets the DDR memory search chip select to an active HIGH.  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSACTIVE? might return SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSACTIVE LOW indicating the chip select is set to an active LOW.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSLevel

This command sets or queries the DDR memory search chip-select level.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSLevel <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSLevel?

**Related Commands** [SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSMode](#)

**Arguments** <NR3> is the amount of chip-select level in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSLEVEL 1.1 sets the chip-select level for DDR memory searches to 1.1 volts.  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSLEVEL? might return SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSLEVEL 0.0000 indicating the chip-select level for DDR memory searches is set to zero volts.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSMode

This command sets or queries the chip-select mode for DDR memory searches.

**Group** Search and Mark



**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSMode {AUTO|MANua1}  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSMode?

**Related Commands** [SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSLevel](#)

**Arguments** AUTO mode calculates the chip-select level for you.  
MANUAL lets you manually set the chip-select level.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSMODE MANUAL sets the chip-select mode for DDR memory searches to MANUAL.  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSMODE? might return  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSMODE AUTO indicating the chip-select mode for DDR memory searches is set to AUTO.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSSource

This command sets or queries the chip select source for DDR memory searches.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSSource  
{CH<x>|MATH<x>|REF<x>|NONE}  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:CSSource?

**Arguments** CH<x> specifies one input channel as the search source, where <x> is the channel number and can be 1, 2, 3, or 4.  
MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.  
REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.  
NONE specifies no CSSource.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSSOURCE REF1 sets the chip select source for DDR memory searches to reference 1.  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSSOURCE? might return  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:CSSOURCE CH1 indicating the chip select source for DDR memory searches is channel 1.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:DATASource

This command sets or queries the data source for a DDR Memory analysis search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:DATASource {CH<x> MATH<x> REF<x>} SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:DATASource?
<b>Arguments</b>	CH<x> specifies one input channel as the search source, where <x> is the channel number.  MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.  REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:DATASOURCE CH2 sets the DDR data source to CH2.  SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:DATASOURCE? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:DATASOURCE CH1, indicating that the DDR data source is set to CH1.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:HYSterisis

This command sets or queries the hysteresis percent for DDR memory searches.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:HYSterisis <NR3> SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:HYSterisis?
<b>Arguments</b>	<NR3> is the hysteresis amount in percent.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:HYSTERISIS 0.10 sets the hysteresis for DDR memory searches to 0.10 percent.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:HYSTERISIS? might return  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:HYSTERISIS 1.0000 indicating  
the hysteresis for DDR memory searches is set to 1.0 percent.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MARgin

This command sets or queries the termination logic margin for back-to-back writes.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MARgin <NR3> SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MARgin?
<b>Arguments</b>	<NR3> is the termination logic margin in percent.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MARGIN 1.0000 sets the termination logic margin to 1.0 percent.  SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MARGIN? might return SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MARGIN 0.0000 indicating the termination logic margin is set to 0.0 percent.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MAXCas

This command sets or queries the maximum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MAXCas <NR3> SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MAXCas?
<b>Arguments</b>	<NR3> is the maximum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MAXCAS 2.0000 sets MAXCas to 2.0 clock periods to the left of the start of the mark.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MAXCAS? might return  
 SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MAXCAS 3.0000 indicating  
 MAXCas is set to 3.0 clock periods to the left of the mark.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MINCas

This command sets or queries the minimum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MINCas <NR3> SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:MINCas?
<b>Arguments</b>	<NR3> is the minimum distance, in clock periods, to the left of the start of the mark where the instrument looks for the chip-select pulse.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MINCAS 2.0000 sets MINCas to 2.0 clock periods to the left of the start of the mark.  SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MINCAS? might return SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:MINCAS 2.0000 indicating MINCas is set to 2.0 clock periods to the left of the mark.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:HIGH

This command sets or queries the value of the data HIGH reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:HIGH <NRF> SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:HIGH?
<b>Related Commands</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTOREflevel
<b>Arguments</b>	<NRF> specifies the value of the data HIGH reference level in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:HIGH 1 sets the data HIGH reference level to 1.0 volt.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:HIGH? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:HIGH 0.0000, indicating that the data HIGH reference level is set to 0 volt.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:LOW

This command sets or queries the value of the data LOW reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:LOW <NRF>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:LOW?

**Related Commands** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTORefLevel

**Arguments** <NRF> specifies the value of the data LOW reference level in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:LOW 0.2 sets the data LOW reference level to 0.2 volt.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:LOW? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:LOW 200.0000E-3, indicating that the data LOW reference level is set to 0.2 volt.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:MID

This command sets or queries the value of the data MID reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:MID <NRF>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATA:MID?

<b>Related Commands</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTORefLevel
<b>Arguments</b>	<NRF> specifies the value of the data MID reference level in volts.
<b>Examples</b>	<p>SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:MID 1.5 sets the data MID reference level to 1.5 volts.</p> <p>SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:MID? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:DATA:MID 1.0000, indicating that the data MID reference level is set to 1.0 volt.</p>

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:HIGH

This command sets or queries the value of the strobe HIGH reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	<pre>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRMemory:REFLevel:STROBE:HIGH &lt;NRF&gt; SEARCH:SEARCH&lt;x&gt;:TRIGger:A:DDRMemory:REFLevel:STROBE:HIGH?</pre>

<b>Related Commands</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTORefLevel
<b>Arguments</b>	<NRF> specifies the value of the strobe HIGH reference level in volts.
<b>Examples</b>	<p>SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:HIGH 5.0 sets the strobe HIGH reference level to 5.0 volts.</p> <p>SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:HIGH? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:HIGH 0.000, indicating that the strobe HIGH reference level is set to 0 volt.</p>

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:LOW

This command sets or queries the value of the strobe LOW reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:LOW  
<NRF>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:LOW?

**Related Commands** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTORefLevel

**Arguments** <NRF> specifies the value of the strobe LOW reference level in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:LOW 1 sets the strobe LOW reference level to 1.0 volt.  
  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:LOW? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:LOW 0.0000, indicating that the strobe LOW reference level is set to 0 volt.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:MID

This command sets or queries the value of the strobe MID reference level for a DDR Memory analysis search to determine where to place a mark. You can set the value only when AUTOREFLEVEL is Custom. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:MID  
<NRF>  
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:MID?

**Related Commands** SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:AUTORefLevel

**Arguments** <NRF> specifies the value of the strobe MID reference level in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:MID 3 sets the strobe MID reference level to 3.0 volts.  
  
SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:MID? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:REFLEVEL:STROBE:MID 0.0000, indicating that the strobe MID reference level is set to 0 volt.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STANdard

This command sets or queries the DDR standard for a DDR Memory analysis search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STANdard {DDR DDR2 DDR3 GDDR3 LPDDR} SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STANdard?
<b>Arguments</b>	<DDR> sets the DDR standard type to DDR. <DDR2> sets the DDR standard type to DDR2. <DDR3> sets the DDR standard type to DDR3. <GDDR3> sets the DDR standard type to GDDR3. <LPDDR> sets the DDR standard type to LPDDR.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STANDARD DDR2 sets the DDR standard to DDR2.  SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STANDARD? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STANDARD DDR3, indicating that the DDR standard is set to DDR3.

## SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STROBEsource

This command sets or queries the strobe source for a DDR Memory analysis search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STROBEsource {CH<x> MATH<x> REF<x>} SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STROBEsource?
<b>Arguments</b>	CH<x> specifies the input channel as the search source, where <x> is the channel number.  MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.



REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.

**Examples** SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STROBESOURCE REF2 sets the DDR strobe source to REF2.

SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STROBESOURCE? might return :SEARCH:SEARCH1:TRIGGER:A:DDRMEMORY:STROBESOURCE CH1, indicating that the DDR strobe source is set to CH1.

## SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:CH<x>

This command sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the channel number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:CH<x>  
{RISe|FALL|EITHer}  
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:CH<x>?

**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:CH2 RISE sets the Channel 2 slope for search 1 to rise.

SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:CH1 RISE, indicating that the slope for the Channel 1 trigger for search 1 is rise.

## SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:MATH<x>

This command sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the math waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:MATH<x>  
 {RISe|FALL|EITHer}  
 SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:MATH<x>?

**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:MATH2 RISe sets the Math 2 waveform slope for search 1 to rise.  
 SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:MATH1? might return  
 :SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:MATH1 RISe, indicating that the slope for the Math 1 waveform trigger for search 1 is rise.

## SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:REF<x>

This command sets or queries the slope for an edge trigger search to determine where to place a mark. <x> is the search number or the reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:REF<x>  
 {RISe|FALL|EITHer}  
 SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:REF<x>?

**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:REF2 RISe sets the Reference 2 waveform slope for search 1 to rise.  
 SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:REF1? might return  
 :SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE:REF1 FALL, indicating that the slope for the Reference 1 waveform trigger for search 1 is the falling edge.

## 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. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce {CH<x> MATH<x> REF<x>} SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOURce?
<b>Arguments</b>	CH<x> specifies one input channel as the edge source, where <x> is the channel number.  MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.  REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.
<b>Examples</b>	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:GLItch:POLarity:CH<x>

This command sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:CH<x> {EITHer NEGAtive POSITIVe} SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:CH<x>?
<b>Arguments</b>	EITHer specifies positive or negative polarity.  NEGAtive specifies a negative polarity.  POSITIVe specifies a positive glitch polarity.

**Examples**     `SSEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:CH2 NEGATIVE`  
specifies that the Channel 2 glitch polarity must be negative for search 1.

`SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:CH1?` might return  
`:SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:CH1 NEGATIVE`,  
indicating that the Channel 1 glitch polarity for search 1 is negative.

## **SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:MATH<x>**

This command sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or math waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:MATH<x>`  
`{EITher|NEGAtive|POSITIve}`  
`SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:MATH<x>?`

**Arguments**     `EITher` specifies positive or negative polarity.  
`NEGAtive` specifies a negative polarity.  
`POSITIve` specifies a positive glitch polarity.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:MATH2 NEGATIVE`  
specifies that the Math 2 waveform glitch polarity must be negative for search 1.

`SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:MATH1?` might return  
`:SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:MATH1 POSITIVE`,  
indicating that the Math 1 waveform glitch polarity for search 1 is positive.

## **SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:REF<x>**

This command sets or queries the polarity for a glitch trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:REF<x>`  
`{EITher|NEGAtive|POSITIve}`  
`SEARCH:SEARCH<x>:TRIGger:A:GLItch:POLarity:REF<x>?`

**Arguments** EITHer specifies positive or negative polarity.  
 NEGAtive specifies a negative polarity.  
 POSITIVE specifies a positive glitch polarity.

**Examples** SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:REF2 NEGATIVE specifies that the Reference 2 waveform glitch polarity must be negative for search 1.  
 SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:GLITCH:POLARITY:REF1 POSITIVE, indicating that the Reference 1 waveform glitch polarity for search 1 is positive.

## SEARCH:SEARCH<x>:TRIGger:A:GLItch:TRIGIF

This command sets or queries the accept or reject state for a glitch trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:GLItch:TRIGIF {ACccept|REJect}  
 SEARCH:SEARCH<x>:TRIGger:A:GLItch:TRIGIF?

**Arguments** ACccept specifies that a glitch occurs.  
 REJect specifies that a glitch does not occur.

**Examples** SEARCH:SEARCH1:TRIGGER:A:GLITCH:TRIGIF ACccept specifies to place a mark if the A trigger glitch occurs.  
 SEARCH:SEARCH1:TRIGGER:A:GLITCH:TRIGIF? might return :SEARCH:SEARCH1:TRIGGER:A:GLITCH:TRIGIF ACccept, indicating that a mark will be placed if a glitch occurs.

## SEARCH:SEARCH<x>:TRIGger:A:GLItch:WIDth

This command sets or queries the glitch width for a glitch trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:GLItch:WIDth <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:GLItch:WIDth?

**Arguments** <NR3> specifies the glitch width in seconds.

**Examples** SEARCH:SEARCH1:TRIGGER:A:GLITCH:WIDTH 0.000000002 sets the search1 glitch width to 2.0 ns.  
  
SEARCH:SEARCH1:TRIGGER:A:GLITCH:WIDTH? might return  
:SEARCH:SEARCH1:TRIGGER:A:GLITCH:WIDTH 1.50E-6, indicating that the search 1 glitch width is set to 1.50  $\mu$ s.

## SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>

This command sets or queries the level for an edge trigger search to determine where to place a mark. <x> is the search number or the channel number. Each channel can have an independent level.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x> {<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>?

**Arguments** <NR3> specifies the trigger level, in volts.  
  
ECL specifies a -1.3 V trigger level.  
  
TTL specifies a 1.4 V trigger level.

**Examples** SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH2 1.0 sets the Channel 1 A trigger level for search 1 to 1.0 Volts.  
  
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1? might return  
:SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1 200.0000E-3, indicating the Channel 1 trigger level for search 1 is 200 mV.

## SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH<x>

This command sets or queries the math waveform level for an edge trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LEVe1:MATH<x> {<NR3> ECL TTL} SEARCH:SEARCH<x>:TRIGger:A:LEVe1:MATH<x>?
<b>Arguments</b>	<NR3> specifies the trigger level in volts. ECL specifies a -1.3 V trigger level. TTL specifies a 1.4 V trigger level.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH2 0.20 sets the Math 2 waveform trigger level for search 1 to 200 mV.  SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH1 100.0000E-3, indicating that the Math 2 waveform trigger level for search 1 is set to 100 mV.

## SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x>

This command sets or queries the specified reference waveform level for a trigger search to determine where to place a mark. SEARCH<x> is the search number and REF<x> is the reference channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x> {<NR3> ECL TTL} SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x>?
<b>Arguments</b>	<NR3> specifies the trigger level in volts. ECL specifies a -1.3 V trigger level. TTL specifies a 1.4 V trigger level.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF2 0.20 sets the Reference 2 waveform trigger level for search 1 to 200 mV.  SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1 200.0000E-3, indicating that the Reference 1 waveform trigger level for search 1 is set to 200 mV.

## 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. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion {AND NAND NOR OR} SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion?
<b>Arguments</b>	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.
<b>Examples</b>	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:CH<x>

This command sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x> {HIGH LOW X} SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x>?
<b>Arguments</b>	HIGH specifies the logic high. LOW specifies the logic low. X specifies a “don’t care” state.



**Examples** SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH2 HIGH sets the logic input for Channel 1 to HIGH.

SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1? might return  
:SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1 LOW, indicating the logic input on Channel 1 for search 1 is set to LOW.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:MATH<x>

This command sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the math waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:MATH<x> {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:MATH<x>?

**Arguments** HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition.

**Examples** SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH2 HIGH sets the Math 2 waveform logic input for search 1 to HIGH.

SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH1? might return  
:SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH1 LOW, indicating the Math 1 waveform logic input for search 1 is set to LOW.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:REF<x>

This command sets or queries the Boolean logic criteria for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the reference channel number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:REF<x> {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:REF<x>?

<b>Arguments</b>	<p>HIGH specifies a high logic level.</p> <p>LOW specifies a low logic level.</p> <p>X specifies a “don’t care” condition.</p>
<b>Examples</b>	<p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF2 HIGH sets the Reference 2 waveform logic input for search 1 to HIGH.</p> <p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1 LOW, indicating the Reference 1 waveform logic input for search 1 is set to LOW.</p>

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>

This command sets or queries the channel threshold level for a pattern or state trigger search to determine where to place a mark. SEARCH<x> is the search number and CH<x> is the channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	<p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:THReshold:CH&lt;x&gt; &lt;NR3&gt;</p> <p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:THReshold:CH&lt;x&gt;?</p>
<b>Arguments</b>	<NR3> specifies the threshold level in volts.
<b>Examples</b>	<p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH2 1.3 sets the Channel 2 logic threshold for search 1 to 1.3 Volts.</p> <p>SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1 2.0000, indicating the Channel 1 logic threshold for search 1 is set to 2.0 Volts.</p>

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH<x>

This command sets or queries the math waveform threshold level for a pattern or state trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
--------------	-----------------

<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LOGic:THResho1d:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:LOGic:THResho1d:MATH<x>?
<b>Arguments</b>	<NR3> specifies the threshold level in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH2 2.0 sets the Math 2 waveform logic trigger threshold for search 1 to 2.0 Volts.  SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH1 1.4000, indicating the Math 1 waveform logic threshold for search 1 is set to 1.4 Volts.

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x>

This command sets or queries the reference waveform threshold level for a pattern or state trigger search to determine where to place a mark. <x> is the search number or the reference waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:LOGic:THResho1d:REF<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:LOGic:THResho1d:REF<x>?
<b>Arguments</b>	<NR3> specifies the threshold level in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF2 2.0 sets the Reference 2 waveform logic trigger threshold for search 1 to 2.0 Volts.  SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1 1.4000, indicating the Reference 1 waveform logic trigger threshold for search 1 is set to 1.4 Volts.

## SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn

This command sets or queries the pattern trigger pattern test that determines where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
--------------	-----------------

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn  
 {TRUE|FALSE|LESSThan|MOREThan}  
 SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn?

**Related Commands** [SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:LESSLimit](#),  
[SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit](#)

**Arguments** TRUE specifies a TRUE logic level.  
 FALSE specifies a FALSE logic level.  
 LESSThan specifies the pattern must be TRUE less than the time limit.  
 MOREThan specifies the pattern must be TRUE more than the time limit.

**Examples** SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN TRUE sets the pattern trigger to trigger when the pattern is TRUE.  
 SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN? might return  
 :SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN TRUE, indicating search 1 will place a mark when the pattern trigger pattern is TRUE.

## SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:LESSLimit

This command sets or queries the time the pattern trigger lesslimit test uses to determines where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:LESSLimit <NR3>  
 SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:LESSLimit?

**Related Commands** [SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn](#), [SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit](#)

**Arguments** <NR3> is the time in seconds.

**Examples** SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:LESSLIMIT 1.0e-9 sets the less than time limit to 1 ns.  
 SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:LESSLIMIT? might return  
 :SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:LESSLIMIT 1.0000E-9, indicating the lesslimit time is set to 1 ns.

## SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit

This command sets or queries the time the pattern trigger morelimit test uses to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit <NR3> SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:MORELimit?
<b>Related Commands</b>	<a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PATtern:WHEn</a> , <a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:PATtern:WHEn:LESSLimit</a>
<b>Arguments</b>	<NR3> is the time in seconds.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:MORELIMIT 2.0e-9 sets the more limit time to 2 ns.  SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:MORELIMIT? might return :SEARCH:SEARCH1:TRIGGER:A:PATTERN:WHEN:MORELIMIT 500.0000E-12, indicating the more limit time is set to 500 ps.

## SEARCH:SEARCH<x>:TRIGger:A:PULse:SOUrce

This command sets or queries the source waveform for a pulse trigger search to determine where to place a mark. <x> is the search number.

---

**NOTE.** *This is the source for Glitch, Width, Runt, Timeout or Window searches.*

---

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:PULse:SOUrce {CH<x> MATH<x> REF<x>} SEARCH:SEARCH<x>:TRIGger:A:PULse:SOUrce?
<b>Arguments</b>	CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.  REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:PULSE:SOURCE REF1` sets the pulse trigger source for search1 to the Reference 1 waveform.

`SEARCH:SEARCH1:TRIGGER:A:PULSE:SOURCE?` might return  
`:SEARCH:SEARCH1:TRIGGER:A:PULSE:SOURCE MATH1`, indicating the pulse trigger source for search 1 is set to the Math 1 waveform.

## **SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity:CH<x>**

This command sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or channel number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity:CH<x>`  
`{EITher|NEGAtive|POSITIve}`  
`SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity:CH<x>?`

**Arguments**     `EITher` places a mark on a runt of either polarity.

`NEGAtive` places a mark when the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.

`POSITIve` places a mark when the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:CH2 NEGATIVE` sets the Channel 2 runt polarity for search 1 to negative.

`SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:CH1?` might return  
`:SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:CH1 POSITIVE`, indicating the Channel 1 runt polarity for search 1 is set to positive.

## **SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity:MATH<x>**

This command sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or math waveform number.

**Group**     Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity:MATH<x>  
 {EITHer|NEGAtive|POSITIVe}  
 SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity:MATH<x>?

**Arguments** EITHer places a mark on a runt of either polarity.  
 NEGAtive places a mark when the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.  
 POSItive places a mark when the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

**Examples** SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:MATH2 NEGATIVE sets the Math 2 waveform runt polarity for search 1 to negative.  
 SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:MATH1 POSITIVE, indicating the Math 1 waveform runt polarity for search 1 is set to positive.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity:REF<x>

This command sets or queries the polarity setting for a runt trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity:REF<x>  
 {EITHer|NEGAtive|POSITIVe}  
 SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLArity:REF<x>?

**Arguments** EITHer places a mark on a runt of either polarity.  
 NEGAtive places a mark when the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.  
 POSItive places a mark when the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

**Examples** SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:REF2 NEGATIVE sets the Reference 1 waveform runt polarity for search 1 to negative.

SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:REF1? might return  
:SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY:REF1 POSITIVE,  
indicating the Reference 1 waveform runt polarity for search 1 is set to positive.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:CH<x>

This command sets or queries the channel threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:CH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGh:CH2 1.3 sets the runt threshold high level to 1.3 Volts.  SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGh:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGh:CH1 2.0000, indicating the Channel 1 runt trigger high threshold is set to 2.0 Volts.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:MATH<x>

This command sets or returns the math waveform threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGh:MATH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold level.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGh:MATH2 1.3 sets the runt trigger HIGH threshold to 1.3 Volts.



SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIG:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIG:MATH1 2.0000, indicating that the Math 1 waveform runt trigger HIGH threshold is set to 2.0 Volts.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIG:REF<x>

This command sets or returns the reference waveform threshold HIGH level for a runt trigger search to determine where to place a mark. <x> is the search number or the reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIG:REF<x> <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIG:REF<x>?

**Arguments** <NR3> specifies the HIGH threshold in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIG:REF2 1.3 sets the runt trigger HIGH threshold to 1.3 Volts.

SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIG:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIG:REF1 2.0000, indicating that the Reference 1 waveform runt trigger HIGH threshold is set to 2.0 Volts.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:CH<x>

This command sets or queries the channel waveform threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the channel number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:CH<x> <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:CH<x>?

**Arguments** <NR3> specifies the LOW threshold in volts.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:CH2 0.10` sets the Channel 2 LOW runt trigger threshold for search 1 to 100 mV.

`SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:CH1?` might return `:SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:CH1 800.0000E-3`, indicating the Channel 1 runt trigger LOW threshold for search 1 is set to 800 mV.

## **SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:MATH<x>**

This command sets or queries the math waveform threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the math waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:MATH<x> <NR3>`  
`SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:MATH<x>?`

**Arguments**     <NR3> specifies the LOW threshold level.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:MATH2 0.10` sets the Math 2 waveform runt trigger LOW threshold for search 1 to 100 mV.

`SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:MATH1?` might return `:SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:MATH1 800.0000E-3`, indicating the Math 1 waveform runt trigger LOW threshold for search 1 is set to 800 mV.

## **SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:REF<x>**

This command sets or returns the reference waveform threshold LOW level for a runt trigger search to determine where to place a mark. <x> is the search number or the reference waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:REF<x> <NR3>`  
`SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW:REF<x>?`

**Arguments**     <NR3> specifies the LOW threshold in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:REF2 0.10 sets the Reference 2 waveform runt trigger LOW threshold for search 1 to 100 mV.

SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW:REF1 800.0000E-3, indicating that the Reference 1 waveform runt trigger LOW threshold for search 1 is 800 mV.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

This command sets or returns the condition setting for a runt trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn {OCCurs|WIDERthan}  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?

**Arguments** OCCurs argument specifies a trigger event if a runt of any detectable width occurs.

WIDERthan argument sets the oscilloscope to trigger if a runt pulse is detected with width wider than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth](#) command.

**Examples** SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEn WIDERTHAN sets the oscilloscope 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. <x> is the search number.

**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. <x> is the search number.

- 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. <x> is the search number.

- Group** Search and Mark
- Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce  
{CH<x>|MATH<x>|REF<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 or 4.  
 MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.  
 REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.

**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:CH<x>

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or channel number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:CH<x>  
 {NR3|ECL|TTL}  
 SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:CH<x>?

**Arguments** <NR3> is the clock level, in volts.  
 ECL specifies a preset ECL high level of -1.3 V.  
 TTL specifies a preset TTL high level of 1.4 V.

**Examples** SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:CH2 ECL sets the Channel 2 setup/hold trigger clock threshold setting for search 1 to ECL.  
 SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:CH1 -1.3000, indicating that the Channel 1 setup/hold trigger clock threshold setting for search 1 is set to -1.3 volts.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:MATH<x>

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:MATH<x> {NR3 ECL TTL} SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:MATH<x>?
<b>Arguments</b>	<NR3> is the clock level, in volts.  ECL specifies a preset ECL high level of -1.3 V.  TTL specifies a preset TTL high level of 1.4 V.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:MATH2 ECL sets the clock threshold setting of the Math 2 waveform for a setup/hold trigger search to ECL.  SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:MATH1 1.4000, indicating that the clock threshold setting of the Math 1 waveform for a setup/hold trigger search is set to 1.4 volts.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:REF<x>

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:REF<x> {NR3 ECL TTL} SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold:REF<x>?
<b>Arguments</b>	<NR3> is the clock level, in volts.  ECL specifies a preset ECL high level of -1.3 V.  TTL specifies a preset TTL high level of 1.4 V.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:REF2 ECL` sets the clock threshold setting of the Reference 2 waveform for a setup/hold trigger search to ECL.

`SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:REF1?` might return `:SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD:REF1 1.0000`, indicating that the clock threshold setting of the Reference 1 waveform for a setup/hold trigger search is set to 1 volt.

## **SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURCE**

This command sets or queries the data source setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURCE`  
                   `{CH<x>|MATH<x>|REF<x>}`  
                   `SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURCE?`

**Arguments**     `CH<x>` specifies an input channel as the search source, where <x> = 1, 2, 3 or 4.

`MATH<x>` specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4.

`REF<x>` specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE MATH1` sets the setup/hold trigger data source setting for search 1 to the Math 1 waveform.

`SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE?` might return `:SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE CH1`, indicating that the setup/hold trigger data source setting for search 1 is set to the Channel 1.

## **SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold:CH<x>**

This command sets or queries the data threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or channel number.

**Group**     Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGGER:A:SETHold:DATA:THReshold:CH<x>  
{NR3|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:DATA:THReshold:CH<x>?

**Arguments** <NR3> is the clock level, in volts.  
ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.

**Examples** SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:CH2 ECL sets the Channel 2 setup/hold trigger data threshold setting for search 1 to -1.3 volts.  
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:CH1 1.4000, indicating that the Channel 1 setup/hold trigger data threshold setting for search 1 is set to 1.4 V.

## SEARCH:SEARCH<x>:TRIGGER:A:SETHold:DATA:THReshold:MATH<x>

This command sets or queries the data threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or math waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGGER:A:SETHold:DATA:THReshold:MATH<x>  
{NR3|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGGER:A:SETHold:DATA:THReshold:MATH<x>?

**Arguments** <NR3> is the clock level, in volts.  
ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.

**Examples** SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:MATH1 ECL sets the Math 2 waveform setup/hold trigger data threshold setting for search 1 to -1.3 volts.  
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:MATH1 1.0000, indicating that the Math 1 waveform setup/hold trigger data threshold setting for search 1 is set to 1.0 volt.



## SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold:REF<x>

This command sets or queries the data threshold setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold:REF<x> {NR3 ECL TTL} SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold:REF<x>?
<b>Arguments</b>	<NR3> is the clock level, in volts.  ECL specifies a preset ECL high level of -1.3 V.  TTL specifies a preset TTL high level of 1.4 V.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:REF2 ECL sets the Reference 2 waveform setup/hold trigger data threshold setting for search 1 to -1.3 volts.  SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD:REF1 1.4000, indicating that the Reference 1 waveform setup/hold trigger data threshold setting for search 1 is set to 1.4 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. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime?
<b>Arguments</b>	<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.
<b>Examples</b>	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:SETTime

This command sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime <NR3> SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime?
<b>Arguments</b>	<NR3> specifies the setup time for setup and hold violation triggering.
<b>Examples</b>	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:CLOCK:EDGE

This command sets or queries the clock slope setting for a state trigger search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:EDGE {FALL RISe} SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:EDGE?
<b>Arguments</b>	FALL specifies polarity as the clock falling edge.  RISe specifies polarity as the clock rising edge.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:EDGE? RISe sets the state trigger clock slope polarity setting for search 1 to RISe.

SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:EDGE? might return  
:SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:EDGE FALL, indicating that  
the state trigger clock slope polarity setting for search 1 is set to FALL.

## SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:SOUrce

This command sets or queries the clock source of the twelve waveforms for a state trigger search to determine where to place a mark. <x> is the search number. The hardware trigger is always on the channel 4.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:SOUrce {CH<x> MATH<x> REF<x>} SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:SOUrce?
<b>Arguments</b>	CH<x> specifies an input channel as the search source, where <x> = 1, 2, 3 or 4. MATH<x> specifies the math waveform as the search source, where <x> = 1, 2, 3 or 4. REF<x> specifies the reference waveform as the search source, where <x> = 1, 2, 3 or 4.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:SOURCE CH2 sets the state trigger clock source for search 1 to Channel 2. SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:SOURCE? might return :SEARCH:SEARCH1:TRIGGER:A:STATE:CLOCK:SOURCE CH1, indicating that the state trigger clock source for search 1 is set to Channel 1.

## SEARCH:SEARCH<x>:TRIGger:A:STATE:WHEN

This command sets or queries the state trigger search to determine where to place a mark. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:STATE:WHEN {TRUE FALSE} SEARCH:SEARCH<x>:TRIGger:A:STATE:WHEN?

- Arguments** TRUE specifies a TRUE logic level.  
FALSE specifies a FALSE logic level.
- Examples** SEARCH:SEARCH1:TRIGGER:A:STATE:WHEN FALSE sets the state trigger logic level for search 1 to FALSE.  
SEARCH:SEARCH1:TRIGGER:A:STATE:WHEN? might return :SEARCH:SEARCH1:TRIGGER:A:STATE:WHEN TRUE, indicating that the state trigger logic level for search 1 is set to TRUE.

## SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity:CH<x>

This command sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or channel number.

- Group** Search and Mark
- Syntax** SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity:CH<x>  
{STAYSHigh|STAYSLOW|EITHer}  
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity:CH<x>?
- Arguments** STAYSHigh specifies the polarity stays HIGH.  
STAYSLOW specifies the polarity stays LOW.  
EITHer specifies the polarity either stays HIGH or stays LOW.
- Examples** SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:CH2 STAYSLOW sets the Channel 2 timeout trigger polarity for search 1 to STAYSLOW.  
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:CH1 STAYSHIGH, indicating that the Channel 2 timeout trigger polarity for search 1 is set to STAYSHIGH.

## SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity:MATH<x>

This command sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or math waveform number.

- Group** Search and Mark

<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLArity:MATH<x> {STAYSHigh STAYSLow EITHer} SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLArity:MATH<x>?
<b>Arguments</b>	STAYSHigh specifies the polarity stays HIGH. STAYSLow specifies the polarity stays LOW. EITHer specifies the polarity either stays HIGH or stays LOW.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:MATH2 STAYSLOW sets the Math 2 waveform timeout trigger polarity for search 1 to STAYSLOW.  SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:MATH1 STAYSHIGH, indicating that the Math 1 waveform timeout trigger polarity for search 1 is set to STAYSHIGH.

## SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLArity:REF<x>

This command sets or queries the polarity for a timeout trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLArity:REF<x> {STAYSHigh STAYSLow EITHer} SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLArity:REF<x>?
<b>Arguments</b>	STAYSHigh specifies the polarity stays HIGH. STAYSLow specifies the polarity stays LOW. EITHer specifies the polarity either stays HIGH or stays LOW.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:REF2 STAYSLOW sets the Reference 2 waveform timeout trigger polarity for search 1 to STAYSLOW.  SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY:REF1 STAYSHIGH, indicating that the Reference 2 waveform timeout trigger polarity for search 1 is set to STAYSHIGH.

## 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. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME <NR3> SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIME?
<b>Arguments</b>	<NR3> is the time in seconds.
<b>Examples</b>	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:TRANSition:DELTATime

This command sets or queries the transition time setting for a transition trigger search to determine where to place a mark.<x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:DELTATime <NR3> SEARCH:SEARCH<x>:TRIGger:A:TRANSition:DELTATime?
<b>Arguments</b>	<NR3> specifies the transition time, in seconds.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the transition trigger time setting for search 1 to 400 ns.  SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME 500.0000E-12, indicating that the transition trigger time setting for search 1 is set to 500 ps.

**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:CH<x>**

This command sets or queries the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:CH<x> {EITHer NEGAtive POSITIve} SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:CH<x>?
<b>Arguments</b>	EITHer specifies either positive or negative polarity.  NEGAtive specifies that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.  POSITIve specifies that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:CH12 NEGATIVE sets the Channel 2 transition trigger polarity setting for search 1 to NEGATIVE.  SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:CH1 POSITIVE, indicating that the Channel 1 transition trigger polarity setting for search 1 is set to POSITIVE.

**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:MATH<x>**

This command sets or queries the polarity setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:MATH<x> {EITHer NEGAtive POSITIve} SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:MATH<x>?
<b>Arguments</b>	EITHer specifies either positive or negative polarity.  NEGAtive specifies that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

**POSITIVE** specifies that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:MATH2 NEGATIVE`  
 sets the Math 2 waveform transition trigger polarity setting for search 1 to `NEGATIVE`.

`SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:MATH1?` might return `:SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:MATH1 POSITIVE`, indicating that the Math 1 waveform transition trigger polarity setting for search 1 is set to `POSITIVE`.

## **SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:REF<x>**

This command sets or queries the polarity setting for a transition trigger search to determine where to place a mark. `<x>` is the search number or reference waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:REF<x>`  
`{EITHer|NEGAtive|POSITIve}`  
`SEARCH:SEARCH<x>:TRIGger:A:TRANSition:POLarity:REF<x>?`

**Arguments**     `EITHer` specifies either positive or negative polarity.

`NEGAtive` specifies that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

`POSITIve` specifies that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:REF2 NEGATIVE`  
 sets the Reference 2 waveform transition trigger polarity setting for search 1 to `NEGATIVE`.

`SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:REF1?` might return `:SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY:REF1 POSITIVE`, indicating that the Reference 2 waveform transition trigger polarity setting for search 1 is set to `POSITIVE`.



**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:CH<x>**

This command sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:CH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:CH2 1.5 sets the Channel 2 transition trigger HIGH threshold setting for search 1 to 1.5 volts.  SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:CH1 2.0000, indicating that the Channel 1 transition trigger HIGH threshold setting for search 1 is set to 2 volts.

**SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:MATH<x>**

This command sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH: MATH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:MATH2 1.5 sets the Math 2 waveform transition trigger HIGH threshold setting for search 1 to 1.5 volts.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:MATH1?  
 might return  
 :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:MATH1  
 2.0000, indicating that the Math 1 waveform transition trigger HIGH threshold  
 setting for search 1 is set to 2 volts.

## SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:REF<x>

This command sets or queries the threshold high voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:REF<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:HIGH:REF<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold in volts.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:REF2 1.5 sets the Reference 2 waveform transition trigger HIGH threshold setting for search 1 to 1.5 volts.  SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH:REF1 2.0000, indicating that the Reference 1 waveform transition trigger HIGH threshold setting for search 1 is set to 2 volts.

## SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:CH<x>

This command sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:CH<x>?

**Arguments** <NR3> specifies the LOW threshold in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:CH2 0.5 sets the Channel 2 transition trigger LOW threshold setting for search 1 to 0.5 volts.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:CH1 800.0000E-3, indicating that the Channel 1 transition trigger LOW threshold setting for search 1 is set to 0.8 volts.

## SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:MATH<x>

This command sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or math waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:MATH<x>  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:MATH<x>?

**Arguments** <NR3> specifies the LOW threshold in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:MATH2 0.5 sets the Math 2 waveform transition trigger LOW threshold setting for search 1 to 0.5 volts.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:MATH1 800.0000E-3, indicating that the Math 1 waveform transition trigger LOW threshold setting for search 1 is set to 0.8 volts.

## SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THReshold:LOW:REF<x>

This command sets or queries the threshold low voltage level setting for a transition trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THResho1d:LOW:REF<x>  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:THResho1d:LOW:REF<x>?

**Arguments** <NR3> specifies the LOW threshold in volts.

**Examples** SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:REF2 0.5  
sets the Reference 2 waveform transition trigger LOW threshold setting for search 1 to 0.5 volts.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:REF1? might  
return :SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW:REF1  
800.0000E-3, indicating that the Reference 1 waveform transition trigger LOW  
threshold setting for search 1 is set to 0.8 volts.

## SEARCH:SEARCH<x>:TRIGger:A:TRANSition:WHEn

This command sets or queries the condition setting for a transition trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:TRANSition:WHEn  
{FASTERthan|SLOWERthan}  
SEARCH:SEARCH<x>:TRIGger:A:TRANSition:WHEn?

**Arguments** FASTERthan sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

**Examples** SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEn SLOWERTHAN sets the  
transition trigger condition setting for search 1 to SLOWERTHAN.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEn? might return  
:SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEn FASTERTHAN, indicating  
that the transition trigger condition setting for search 1 is set to FASTERTHAN.

## 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. <x> is the search number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	<pre>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TYPE {EDGE RUNT TRANSition PATtern GLItch SETHold UNDEFINED  WIDth TIMEOut WINDow STATE DDRRead DDRWrite DDRREADWrite} SEARCH:SEARCH&lt;x&gt;:TRIGger:A:TYPE?</pre>
<b>Arguments</b>	<p>EDGE triggers when the source input signal amplitude crosses the specified level in the direction given by the slope.</p> <hr/> <p><b>NOTE.</b> <i>EDGE is standard. All other arguments are optional.</i></p> <hr/> <p>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&lt;x&gt;:TRIGger:A:RUNt:HIGH and SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNt:LOW THRESHOLD commands.</p> <p>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.</p> <p>PATtern trigger on a functional pattern combination of one to four input signal sources.</p> <p>GLItch triggers at the trailing edge of the glitch pulse where the signal crosses the trigger level.</p> <p>SETHold triggers on the data input signal transitions relative to the clock input signal transitions within the setup and hold time violation zone.</p> <p>UNDEFINED is the default. To perform a search, select a different argument.</p> <p>WIDth triggers on input signal source pulses that are inside or outside of the given time range.</p> <p>TIMEOut triggers on an input signal source that stays above, stays below, or stays either above or below the trigger level for a given time.</p> <p>WINDow triggers on an input signal source that enters or exits the horizontal band defined by the two trigger levels.</p> <p>STATE triggers on a functional pattern combination of one to three data sources at the time of the clock transition.</p> <p>DDRRead triggers at the start and end of the read bus.</p> <p>DDRWrite triggers at the start and end of the write bus.</p> <p>DDRREADWrite triggers at the start and end of both the read and write bus.</p>

**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:SEARCH<x>:TRIGger:A:WIDth:HIGHLimit

This command sets or queries the upper time limit setting for a width trigger search to determine where to place a mark. <x> is the search number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:WIDth:HIGHLimit <NR3>`  
`SEARCH:SEARCH<x>:TRIGger:A:WIDth:HIGHLimit?`

**Arguments**     <NR3> specifies the upper time limit in seconds.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:WIDTH:HIGHLIMIT 400E-12` sets the width trigger upper time limit setting for search 1 to 0.4 ns.

`SEARCH:SEARCH1:TRIGGER:A:WIDTH:HIGHLIMIT?` might return  
`:SEARCH:SEARCH1:TRIGGER:A:WIDTH:HIGHLIMIT 500.0000E-12`,  
 indicating that the width trigger upper time limit setting for search 1 is set to 0.5 ns.

## SEARCH:SEARCH<x>:TRIGger:A:WIDth:LOWLimit

This command sets or queries the lower time limit setting for a width trigger search to determine where to place a mark. <x> is the search number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:WIDth:LOWLimit <NR3>`  
`SEARCH:SEARCH<x>:TRIGger:A:WIDth:LOWLimit?`

**Arguments**     <NR3> specifies the lower time limit in seconds.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:WIDTH:LOWLIMIT 400E-12` sets the width trigger lower time limit setting for search 1 to 0.4 ns.

SEARCH:SEARCH1:TRIGGER:A:WIDTH:LOWLIMIT? might return  
:SEARCH:SEARCH1:TRIGGER:A:WIDTH:LOWLIMIT 500.0000E-12,  
indicating that the width trigger lower time limit setting for search 1 is set to 0.5 ns.

## SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:CH<x>

This command sets or queries the polarity for a width trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:CH<x> {NEGAtive POSITIVe} SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:CH<x>?
<b>Arguments</b>	NEGAtive specifies a negative polarity. POSITIVe specifies a positive polarity.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:CH2 NEGATIVE sets the Channel 2 width trigger polarity setting for search 1 to NEGATIVE.  SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:CH1 POSITIVE, indicating that the Channel 1 width trigger polarity setting for search 1 is set to POSITIVE .

## SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:MATH<x>

This command sets or queries the polarity for a width trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:MATH<x> {NEGAtive POSITIVe} SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLarity:MATH<x>?
<b>Arguments</b>	NEGAtive specifies a negative polarity. POSITIVe specifies a positive polarity.

**Examples** SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:MATH2 NEGATIVE sets the Math 2 waveform width trigger polarity setting for search 1 to NEGATIVE.

SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:MATH1 POSITIVE, indicating that the Math 1 waveform width trigger polarity setting for search 1 is set to POSITIVE.

## SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLArity:REF<x>

This command sets or queries the polarity for a width trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLArity:REF<x>  
{NEGAtive|POSITIve}  
SEARCH:SEARCH<x>:TRIGger:A:WIDth:POLArity:REF<x>?

**Arguments** NEGAtive specifies a negative polarity.  
POSITIve specifies a positive polarity.

**Examples** SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:REF2 NEGATIVE sets the Reference 2 waveform width trigger polarity setting for search 1 to NEGATIVE.

SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:WIDTH:POLARITY:REF1 POSITIVE, indicating that the Reference 1 waveform width trigger polarity setting for search 1 is set to POSITIVE.

## SEARCH:SEARCH<x>:TRIGger:A:WIDth:WHEn

This command sets or queries the width trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:WIDth:WHEn {OUTSide|WIDthin}  
SEARCH:SEARCH<x>:TRIGger:A:WIDth:WHEn?



**Arguments**    **OUTside** specifies the outside limits.  
**WITHin** specifies the inside limits.

**Examples**    **SEARCH:SEARCH1:TRIGGER:A:WIDTH:WHEN OUTSIDE** sets the width trigger for search 1 to outside limits.  
**SEARCH:SEARCH1:TRIGGER:A:WIDTH:WHEN?** might return  
**:SEARCH:SEARCH1:TRIGGER:A:WIDTH:WHEN WITHIN**, indicating that the width trigger for search 1 is set to inside limits.

## **SEARCH:SEARCH<x>:TRIGger:A:WINDow:EVENT**

This command sets or queries the window trigger search to determine where to place a mark. <x> is the search number.

**Group**    Search and Mark

**Syntax**    **SEARCH:SEARCH<x>:TRIGger:A:WINDow:EVENT**  
**{OUTSIDEGreater|INSIDEGreater|ENTERSWindow|EXITSWindow}**  
**SEARCH:SEARCH<x>:TRIGger:A:WINDow:EVENT?**

**Arguments**    **OUTSIDEGreater** specifies a trigger event when the signal leaves the window defined by the threshold levels for the time specified by **Width**.

**INSIDEGreater** specifies a trigger event when the signal enters the window defined by the threshold levels for the time specified by **Width**.

**ENTERSWindow** specifies a trigger event when the signal enters the window defined by the threshold levels.

**EXITSWindow** specifies a trigger event when the signal leaves the window defined by the threshold levels.

**Examples**    **SEARCH:SEARCH1:TRIGGER:A:WIDTH:EVENT OUTSIDEGREATER** sets the window trigger for search 1 specifies a trigger when the signal leaves the window defined by the threshold levels for the time specified by **width**.

**SEARCH:SEARCH1:TRIGGER:A:WIDTH:EVENT?** might return  
**:SEARCH:SEARCH1:TRIGGER:A:WIDTH:EVENT ENTERSWINDOW**, indicating that the window trigger for search 1 specifies a trigger when the signal enters the window defined by the threshold levels.

## SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:CH<x>

This command sets or queries the channel threshold HIGH level for a window trigger search to determine where to place a mark. <x> is the search number or channel number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:CH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:CH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold level.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:CH2 1.5000 sets the Channel 2 window trigger high threshold level for search 1 to 1.5 volts.  SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:CH1? might return :SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:CH1 2.0000, indicating that the Channel 2 window trigger high threshold level for search 1 is set to 2.0 V.

## SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:MATH<x>

This command sets or queries the math waveform threshold HIGH level for a window trigger search to determine where to place a mark. <x> is the search number or math waveform number.

<b>Group</b>	Search and Mark
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:MATH<x> <NR3> SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:MATH<x>?
<b>Arguments</b>	<NR3> specifies the HIGH threshold level.
<b>Examples</b>	SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:MATH2 1.5000 sets the Math 2 waveform window trigger high threshold level for search 1 to 1.5 volts.  SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:MATH1? might return :SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:MATH1

2.0000, indicating that the Math 1 waveform window trigger high threshold level for search 1 is set to 2.0 volts.

## SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:REF<x>

This command sets or queries the reference waveform threshold HIGH level for a window trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:REF<x>  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:HIGH:REF<x>?

**Arguments** <NR3> specifies the HIGH threshold level.

**Examples** SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:REF2 1.5000 sets the Reference 2 waveform window trigger high threshold level for search 1 to 1.5 volts.

SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH:REF1 2.0000, indicating that the Reference 1 waveform window trigger high threshold level for search 1 is set to 2.0 volts.

## SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:CH<x>

This command sets or queries the channel waveform threshold LOW level for a window trigger search to determine where to place a mark. <x> is the search number or channel waveform number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:CH<x> <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:CH<x>?

**Arguments** <NR3> specifies the LOW threshold level.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:CH2 0.5` sets the Channel 2 window trigger low threshold level for search 1 to 0.5 volts.

`SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:CH1?` might return `:SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:CH1 800.0000E-3`, indicating that the Channel 1 window trigger high threshold level for search 1 is set to 0.8 volts.

## **SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:MATH<x>**

This command sets or queries the math waveform threshold LOW level for a window trigger search to determine where to place a mark. <x> is the search number or math waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:MATH<x>`  
                   `<NR3>`  
                   `SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:MATH<x>?`

**Arguments**     <NR3> specifies the LOW threshold level.

**Examples**     `SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:MATH2 0.5` sets the Math 2 waveform window trigger low threshold level for search 1 to 0.5 volts.

`SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:MATH1?` might return `:SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:MATH1 800.0000E-3`, indicating that the Math 1 waveform window trigger low threshold level for search 1 is set to 0.8 volts.

## **SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:REF<x>**

This command sets or queries the reference waveform threshold LOW level for a window trigger search to determine where to place a mark. <x> is the search number or reference waveform number.

**Group**     Search and Mark

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:REF<x> <NR3>`  
                   `SEARCH:SEARCH<x>:TRIGger:A:WINDow:THReshold:LOW:REF<x>?`

**Arguments** <NR3> specifies the LOW threshold level.

**Examples** SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:REF2 0.5 sets the Reference 2 waveform window trigger low threshold level for search 1 to 0.5 volts.

SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:REF1? might return :SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW:REF1 2.0000, indicating that the Reference 1 waveform window trigger low threshold level for search 1 is set to 0.8 volts.

## SEARCH:SEARCH<x>:TRIGger:A:WINDow:WIDth

This command sets or queries the width setting for a window trigger search to determine where to place a mark. <x> is the search number.

**Group** Search and Mark

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:WINDow:WIDth <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:WINDow:WIDth?

**Arguments** <NR3> specifies the minimum width, in seconds.

ECL specifies a preset high level of -1.3V and TTL specifies a preset high level of +1.4V.

**Examples** SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 400E-12 sets the window trigger width setting for search 1 to 0.4 ns.

SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH? might return :SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 500.0000E-12, indicating that the window trigger width setting for search 1 is set to 0.5 ns.

## SEARCH:STOP

This command sets or queries the stop of acquisitions when a search finds an event.

**Group** Search and Mark

**Syntax** SEARCH:STOP {<NR1>|OFF|ON|1|0}  
SEARCH:STOP?

**Arguments** <NR1> any numerical nonzero, stops the acquisition.  
 OFF continues the acquisition.  
 ON stops the acquisition.  
 1 stops the acquisition.  
 0 continues the acquisition.

**Examples** SEARCH:STOP 1 stops the acquisition.  
 SEARCH:STOP? might return :SEARCH:STOP 1, indicating that a search found an event and stops acquisition.

## SElect? (Query Only)

This query-only command returns the selected waveform that is affected by the front panel controls and the display status (on or off) of all waveforms. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Source waveform setting.

**Group** Vertical

**Syntax** SElect?

**Examples** SELECT? might return the waveform and status information as :SELECT:CH1 1;CH2 0;CH3 0;CH4 0;MATH1 0; MATH2 0;MATH3 0;MATH4 0;REF1 0;REF2 0; REF3 0;REF4 0;CONTROL CH1.

## SElect:B<x>

This command sets or queries the display state for the bus specified by x. The value of x can range from 1 through 16.

**Group** Bus

**Syntax** SElect:B<x> {<NR1>|OFF|ON}  
 SElect:B<x>?

**Related Commands** [SElect:CH<x>](#), [SElect:D<x>](#), [SElect:<wfm>](#)

**Arguments** <NR1> = 0 turns off the display of the specified bus; any other value turns on the display of the specified bus.

OFF turns off the display of the indicated bus.

ON turns on the display of the indicated bus.

**Examples** SELECT:B1 1 displays bus B1.

SELECT:B1? might return SELECT:B1 ON, indicating bus B1 is displayed.

## SElect:CH<x>

This command sets or queries the displayed state of the specified channel waveform. The x can be channel 1 through 4.

**Group** Vertical

**Syntax** SElect:CH<x> {<NR1>|OFF|ON}  
SElect:CH<x>?

**Related Commands** [SElect:B<x>](#), [SElect:D<x>](#), [SElect:<wfm>](#)

**Arguments** <NR1> = 0 turns off the display of the specified channel waveform; any other value turns on the display of the specified waveform.

OFF turns off the display of the indicated channel waveform.

ON displays the indicated channel waveform.

**Examples** SELECT:CH1 ON displays Channel 1.

SELECT:CH1? might return SELECT:CH1 1, indicating Channel 1 is displayed.

## SElect:CONTROI <wfm>

This command sets or queries the waveform that is selected as the implied recipient of channel-related commands that support legacy-style programs. The command form also performs the equivalent of a [SElect:<wfm>](#) ON command. This command is equivalent to selecting Measurement Setup from the Measure menu and either viewing or setting the Source waveform.

**Group** Vertical

**Syntax**     `SElect:CONTROL {CH<x>|MATH<x>|REF<x>}  
SElect:CONTROL?`

**Arguments**     `CH<x>` selects the specified channel waveform as the waveform that is affected by the front panel controls. The x variable can be expressed as an integer ranging from 1 through 4.

`MATH<x>` selects the specified math waveform as the waveform that is affected by the front panel controls. The x variable can be expressed as an integer ranging from 1 through 4.

`REF<x>` selects the specified reference waveform as the waveform that is affected by the front panel controls. The x variable can be expressed as an integer ranging from 1 through 4.

**Examples**     `SELECT:CONTROLCH2` resets acquisition displays on Channel 2, and causes the selected waveform to be the implied object of waveform commands.

`SELECT:CONTROL?` might return `:SELECT:CONTROL MATH3`, indicating that Math 3 is the implied object of waveform commands.

## SElect:DALL (No Query Form)

This command sets the displayed state of all the digital inputs.

**Group**     Digital

**Syntax**     `SElect:DALL <NR1>`

**Arguments**     `<NR1> = 0` turns off the display of all the digital inputs; any other value turns on the display of all the digital inputs.

`OFF` turns off the display of all the digital inputs.

`ON` displays all the digital inputs.

**Examples**     `SELECT:DALL ON` sets the displayed state of all the digital inputs to on.

## SElect:DIGTraces:COMbination (No Query Form)

This command turns on the digital channels that have binary digits as 1. The binary digits are obtained from the set decimal value.



<b>Group</b>	Digital
<b>Syntax</b>	SElect:DIGTraces:COMBination <nr1>
<b>Arguments</b>	nr1 is a decimal integer.
<b>Examples</b>	SELECT:DIGTRACES:COMBINATION 6 turns on the D1 and D2 digital channels.

## SElect:DIGTraces:LISt

This command turns on the specified digital channels or returns the list of digital channels that are on.

<b>Group</b>	Digital
<b>Syntax</b>	SElect:DIGTraces:LISt <Dx> SElect:DIGTraces:LISt?
<b>Arguments</b>	<Dx> are the specified digital channels, for example D3, D6, D8.
<b>Examples</b>	SELECT:DIGTRACES:LIST D0 D1 turns on the digital channel D0 and D1. SELECT:DIGTRACES:LIST? might return SELECT:DIGTRACES:LIST D3, D4, D5, indicating that digital channels D3, D4, and D5 are on. The other digital channels are off.

## SElect:D<x>

This command sets or queries the display state for the digital input specified by x. The value of x can range from 0 through 15.

<b>Group</b>	Digital
<b>Syntax</b>	SElect:D<x> {<NR1> OFF ON} SElect:D<x>?
<b>Related Commands</b>	<a href="#">SElect:B&lt;x&gt;</a> , <a href="#">SElect:CH&lt;x&gt;</a> , <a href="#">SElect:&lt;wfm&gt;</a>

**Arguments** <NR1> = 0 turns off the display of the specified digital input; any other value turns on the display of the specified bus.  
 OFF turns off the display of the indicated bus.  
 ON turns on the display of the indicated bus.

**Examples** SELECT:D1 ON displays digital input D1.  
 SELECT:D1? might return SELECT:D1 1, indicating digital input D1 is displayed.

## SElect:<wfm>

This command turns on the display of a specified waveform and also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform. WFM can be a channel, math, or reference waveform.

**Group** Vertical

**Syntax** SElect:<wfm> {ON|OFF|<NR1>}  
 SElect:<wfm>?

**Related Commands** [SElect:B<x>](#), [SElect:CH<x>](#), [SElect:D<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.  
 <NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:CH2 ON turns the channel 2 waveform display on, and selects Channel 2.  
 SELECT:REF1? might return :SELECT:REF1 1, indicating that Reference 1 waveform is being displayed.

## SET? (Query Only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET?

query. The SET? query always returns command headers, regardless of the setting of the **HEADer** command. This is because the returned commands are intended to be sent back to the instrument as a command string. The **VERBose** command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the **\*LRN?** command.

**Group** Miscellaneous

**Syntax** SET?

**Related Commands** **HEADer**, **\*LRN?**, **VERBose**

**Examples** SET? might return the following response: :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE NOTRUNNING;:AUXOUT:SOURCE ATRIGGER;EDGE FALLING;:CMD BATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;:DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST 500.0000E-3;:HARDCOPY:FILENAME "untitled";PORT FILE;:DIAG:LEVEL SUBSYS;:SAVE:WAVEFORM:FILEFORMAT INTERNAL;: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;.

## SETUp:NAME

This command allows you to create (or query) a name for your saved setups. The default name for all user setups is “User.” The default name for factory setups is “Factory.”

This command is equivalent to selecting Save As from the File menu, pressing the Setup button, selecting the desired setup location, clicking the keyboard icon, and entering your setup name.

**Group** Save and Recall

**Syntax** SETUP:NAME <NR1>, <QString>  
SETUP:NAME? <NR1>

**Related Commands** [SAVe:SETUp](#)

**Arguments** <NR1> specifies a location in which the setup label is stored. Location values range from 1 through 10.  
<QString> is a string containing the setup label.

**Examples** SETUP:NAME1, "My Setup" changes the setup label for internal setup location 1 to "My Setup".  
SETUP:NAME? 2 might return :SETUP:NAME 2, "My Setup", indicating that "My Setup" is the setup label that you defined for internal setup location 2.

## \*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.

## TEKLink:CONNECTION? (Query Only)

This command returns the current TekLink network connection.

**Group** TekLink

**Syntax** TEKLink:CONNECTION?

**Returns** INSTRUMENT when two instruments are connected through a TekLink cable.  
 HUB when two or more instruments are connected to a hub through a TekLink cable.  
 NONE when the instrument is not connected through a TekLink cable.

**Examples** TEKLink:CONNECTION? might return :TEKLink:CONNECTION:INSTRUMENT, indicating that the two instruments are connected through a TekLink cable.

## TEKLink:REFCik

This command sets or queries the current state of the instrument's TekLink Reference Output.

<b>Group</b>	TekLink
<b>Syntax</b>	TEKLink:REFClk {ON OFF PENDING} TEKLink:REFClk?
<b>Arguments</b>	ON specifies the on state of the instrument's TekLink Reference Output. OFF specifies the off state of the instrument's TekLink Reference Output. PENDING specifies the pending state of the instrument's TekLink Reference Output.
<b>Examples</b>	TEKLINK:REFCLK ON sets the on state of the instrument's TekLink Reference Output. TEKLINK:REFCLK? might return :TEKLINK:REFCLK OFF, indicating that the off state of the instrument's TekLink Reference Output.

## TEKSecure (No Query Form)

This command initializes both waveform and setup memories, overwriting any previously stored data.

TEKSecure deletes all four waveform reference memory slots on the hard drive, if they exist, and puts all setups in the factory-initialized state. External setups that are stored on the hard drive are not affected.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	TEKSecure
<b>Examples</b>	TEKSECURE initializes both waveform and setup memories.

## TEST

This command provides the ability to select and execute an item at any level of the test hierarchy (Test, Area or Subsystem). The query returns the last command sent. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, choosing a test and then pressing Run.

---

**NOTE.** We recommend turning off both *DIAG:CONTROL:HALT* and *DIAG:CONTROL:LOOP* before executing the *TEST* command.

---

<b>Group</b>	Diagnostics
<b>Syntax</b>	TEST <QString> TEST?
<b>Related Commands</b>	<a href="#">TEST:RESults?</a> , <a href="#">TEST:RESults:VERBose?</a> , <a href="#">TEST:STOP</a>
<b>Arguments</b>	<QString> sets the test ID, which ranges from 0 through 3 characters. If no test ID is specified, all available diagnostics are executed.
<b>Examples</b>	<p>TEST"1" executes all Acquisition tests (Subsystem:Acquisition).</p> <p>TEST"11" executes all Acquisition Memory tests (Subsystem:Acquisition, Area:Memory).</p> <p>TEST"113" executes a specific Acquisition Memory test (Subsystem:Acquisition, Area:Memory, Test:Address Lines).</p>

## TEST:RESults? (Query Only)

This query-only command returns an abbreviated status about the results of the last TEST execution. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

<b>Group</b>	Diagnostics
<b>Syntax</b>	TEST:RESults?
<b>Related Commands</b>	<a href="#">TEST</a> , <a href="#">TEST:RESults:VERBose?</a>
<b>Examples</b>	<p>TEST:RESULTS? might return :TEST:RESULTS "135", indicating the specific failure code of the first test failure.</p> <p>TEST:RESULTS? might return :TEST:RESULTS "PASS", indicating that the previously executed test passed.</p> <p>TEST:RESULTS? might return :TEST:RESULTS "*****", indicating that the test was not run.</p>

## TEST:RESults:VERBoSe? (Query Only)

This query-only command returns a more explanatory message about the results of the last TEST execution than the [TEST:RESults?](#) query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking the Error Log button.

**Group**     Diagnostics

**Syntax**    TEST:RESults:VERBoSe?

**Related Commands**    [TEST](#), [TEST:RESults?](#)

**Examples**     TEST:RESULTS:VERBOSE? might return :TEST:RESULTS:VERBOSE DIAG ABORTED, ERROR 15 in A2D-Dmux 1 interconnects".

## TEST:STOP (No Query Form)

This command (no query form) causes test (or diagnostics) execution to terminate at the end of the next low-level test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking Abort.

**Group**     Diagnostics

**Syntax**    TEST:STOP

**Related Commands**    [DIAG:STOP](#)

**Examples**     TEST:STOP terminates test (or diagnostics) execution at the end of the next low-level test.

## TIME

This command sets or queries the time that the instrument displays. This command is equivalent to selecting Set Time & Date from the Utilities menu and then setting the fields in the Time group box.

**Group**     Miscellaneous



**Syntax** TIME <QString>  
TIME?

**Related Commands** DATE

**Arguments** <QString> is a 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.

**Examples** TIME“14:00:00” sets the time to exactly 2:00 p.m.  
TIME? might return :TIME "14:05:17, indicating the current time is set to 2:05 p.m. and 17 seconds.

## \*TRG (No Query Form)

This command (no query form) performs the 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 selecting A Event (Main) Trigger Setup from the Trig menu and then clicking Force Trigger.

**Examples** TRIGGERFORCE forces a trigger event to occur.

```

TRIGGER? might return these trigger parameters: :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;:TRIGGER:A:LOGIC:CLASS PATTERN;FUNCTION
AND;THRESHOLD:CH1 1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;:TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2
X;CH3 X;:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4
X;:TRIGGER:A:LOGIC:PATTERN:WHEN
TRUE;WHEN:LESSLIMIT 5.0000E-9;MORELIMIT
5.0000E-9;:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD
1.2000;SOURCE CH2;:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD
1.2000;SOURCE CH1;:TRIGGER:A:LOGIC:SETHOLD:HOLDTIME
2.0000E-9;SETTIME 3.0000E-9;:TRIGGER:A:LOGIC:STATE:INPUT:CH4
RISE;:TRIGGER:A:LOGIC:STATE:WHEN
TRUE;:TRIGGER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;POLARITY
POSITIVE;:TRIGGER:A:PULSE:WINDOW:TYPE INSIDE;WHEN
OCCURS;WIDTH 2.0000E-9;LOGIC:INPUT:CH1 HIGH;CH2 HIGH;CH3
HIGH;CH4 HIGH;:TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;:TRIGGER:A:PULSE:RUNT:POLARITY
POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;:TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH
2.0000E-9;:TRIGGER:A:PULSE:TRANSITION:DELTATIME
2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;:TRIGGER:A:PULSE:TRANSITION:WHEN
FASTERTHAN;:TRIGGER:A:PULSE:WIDTH:LOWLIMIT
2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;POLARITY
POSITIVE;:TRIGGER:A:PULSE:TIMEOUT:POLARITY
STAYSHIGH;TIME 2.0000E-9;:TRIGGER:A:VIDEO:CUSTOM:FORMAT
INTERLACED;SCAN RATE1;:TRIGGER:A:VIDEO:FIELD
ALLFIELDS;HOLDOFF:FIELD 0.0000;:TRIGGER:A:VIDEO:LINE
1;POLARITY NORMAL;SCAN RATE1;SOURCE CH1;STANDARD
NTSC;:TRIGGER:A:COMMUNICATION:STANDARD CUSTOM;CODE
NRZ;BITRATE 1544000;AMI:PULSEFORM PLUSONE;THRESHOLD:HIGH
0.0000;LOW 0.0000;:TRIGGER:A:COMMUNICATION:CLOCK:POLARITY
RISE;:TRIGGER:A:COMMUNICATION:CMI:PULSEFORM

```

```
PLUSONE;:TRIGGER:A:COMMUNICATION:SOURCE CH1;SOURCE:TYPE
DATA;:TRIGGER:B:STATE 0;TYPE EDGE;LEVEL 0.0000;BY
EVENTS;EDGE:SOURCE CH1;SLOPE RISE;COUPLING
DC;:TRIGGER:B:TIME 16.0000E-9;EVENTS:COUNT 2.
```

## TRIGger:{A|B}:EDGE? (Query Only)

This query-only command returns the trigger source, coupling, and slope for the specified edge trigger. This command is equivalent to selecting Edge Setup from the Trig menu and viewing the current setups, or selecting B Event (Delayed) Trigger Setup from the Trig menu and viewing the current Source, Slope, and Coupling settings.

**Group** Trigger

**Syntax** TRIGger:{A|B}:EDGE?

**Related Commands** [TRIGger:{A|B}:PULse?](#), [TRIGger:{A|B}:LOGic?](#), [TRIGger:{A|B}:EDGE:COUPling](#), [TRIGger:{A|B}:EDGE:SLOpe](#), [TRIGger:{A|B}:EDGE:SOUrce](#), [TRIGger:B:STATE](#)

**Examples** TRIGGER:A:EDGE? might return :TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC; SLOPE RISE, indicating the trigger source, coupling, and slope for the A edge trigger.

## TRIGger:{A|B}:EDGE:COUPling

This command sets or queries the type of coupling for the edge trigger. This command is equivalent to selecting Event Trigger Setup from the Trig menu, selecting Edge Trigger, and choosing from the Coupling drop-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:EDGE:COUPling  
{AC|DC|HFrej|LFrej|NOISerej|ATRIGger}  
TRIGger:{A|B}:EDGE:COUPling?

**Related Commands** [TRIGger:{A|B}:EDGE:SOUrce](#), [TRIGger:{A|B}:EDGE:SLOpe](#)

<b>Arguments</b>	<p>AC selects AC trigger coupling, which passes the input signals above 60 Hz to the trigger circuitry.</p> <p>DC selects DC trigger coupling, which passes all input signals to the trigger circuitry.</p> <p>HFRej coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.</p> <p>LFRej coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.</p> <p>NOISErej coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may require greater trigger signal amplitude.</p> <p>ATRIGGER this B trigger command sets the B trigger coupling to match the setting on the A trigger.</p>
<b>Examples</b>	<p>TRIGGER:A:EDGE:COUPLING DC sets the A edge trigger coupling to DC.</p> <p>TRIGGER:A:EDGE:COUPLING? might return :TRIGGER:A:EDGE:COUPLING DC, indicating that the A edge trigger coupling is set to DC.</p>

## TRIGger:{A|B}:EDGE:COUPLing:CH<x>

This command sets or queries the type of coupling for the A or B trigger for the specified channel. This command is equivalent to selecting A or B Trigger Setup from the Trig menu and choosing the setting from the Coupling drop-down list.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:{A B}:EDGE:COUPLing:CH&lt;x&gt; {AC DC HFRej LFRej NOISErej} TRIGger:{A B}:EDGE:COUPLing:CH&lt;x&gt;?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:EDGE:COUPLing</a>

<b>Arguments</b>	<p>AC selects AC trigger coupling.</p> <p>DC selects DC trigger coupling.</p> <p>HFRej selects high frequency low sensitivity.</p> <p>LFRej selects low frequency low sensitivity.</p> <p>NOISErej selects DC low sensitivity.</p>
------------------	--

**Examples** TRIGGER:B:EDGE:COUPLING:CH1 DC selects DC for the B trigger coupling of Channel 1.

TRIGGER:B:EDGE:COUPLING:CH1? might return  
:TRIGGER:B:EDGE:COUPLING:CH1 LFREJ for the B trigger coupling.

## TRIGger:{A|B}:EDGE:SLOpe

This command sets or queries the slope for the edge trigger. This command is equivalent to selecting Event Trigger Setup from the Trig menu and then choosing the desired Slope.

**Group** Trigger

**Syntax** TRIGger:{A|B}:EDGE:SLOpe {RISE|FALL|EITHer}  
TRIGger:{A|B}:EDGE:SLOpe?

**Related Commands** [TRIGger:{A|B}:EDGE:SOUrce](#), [TRIGger:{A|B}:EDGE:COUPling](#), [TRIGger:{A|B}:EDGE?](#), [TRIGger:B:STATE](#)

**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 specified 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|B}:EDGE:SLOpe:AUX

This command sets or queries the slope for the edge trigger for the AUX In input. This command is equivalent to selecting A Event or B Event Trigger Setup from the Trig menu and then choosing the desired Slope.

**Group** Trigger

**Syntax** TRIGger:{A|B}:EDGE:SLOpe:AUX {RISe|FALL}  
TRIGger:{A|B}:EDGE:SLOpe:AUX?

- 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.
- Examples** TRIGGER:A:EDGE:SLOPE:AUX RISE sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.  
 TRIGGER:A:EDGE:SLOPE:AUX? might return :TRIGGER:A:EDGE:SLOPE:AUX FALL, indicating that the A edge trigger slope is negative.

## TRIGger:{A|B}:EDGE:SLOpe:CH<x>

This command sets or queries the slope for the edge trigger for the specified channel. This command is equivalent to selecting A Event or B Event Trigger Setup from the Trig menu and then choosing the desired Slope.

- Group** Trigger
- Syntax** TRIGger:{A|B}:EDGE:SLOpe:CH<x> {RISe|FALL|EITHer}  
 TRIGger:{A|B}:EDGE:SLOpe:CH<x>?
- Related Commands** [TRIGger:{A|B}:EDGE:SOUrce](#), [TRIGger:{A|B}: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:CH2 RISE sets the Channel 2 edge trigger slope to rising or positive  
 TRIGGER:A:EDGE:SLOPE:CH1? might return :TRIGGER:A:EDGE:SLOPE FALL, indicating that the A edge trigger slope is negative.

## TRIGger:{A|B}:EDGE:SOUrce

This command sets or queries the source for the edge trigger. This command is equivalent to selecting Event Trigger Setup from the Trig menu and then choosing from the Source drop-down list.

- Group** Trigger

**Syntax** TRIGGER:{A|B}:EDGE:SOURCE {AUXiliary|CH<x>|LINE|D<x>}  
 TRIGGER:{A|B}:EDGE:SOURCE?

**Related Commands** TRIGGER:{A|B}:EDGE:SLOpe, TRIGGER:{A|B}:EDGE:COUPLing,  
 TRIGGER:{A|B}:EDGE?, TRIGGER:B:STATE

**Arguments** AUXiliary specifies an external trigger using the Auxiliary Trigger Input. x can be 1, 2, 3, or 4.  
 CH<x> specifies one input channel as the edge trigger source.  
 LINE specifies AC line voltage, and is for A Trigger only.  
 D<x> specifies a digital input as the edge trigger source. x can be 0 through 15.

**Examples** TRIGGER:A:EDGE:SOURCE CH2 sets the A edge trigger source to input channel CH2.  
 TRIGGER:A:EDGE:SOURCE? might return :TRIGGER:A:EDGE:SOURCE CH1, indicating that the A edge trigger source is set to input channel CH1.

## TRIGGER:{A|B}:LEVEL

This command sets or queries the level for the trigger. This command is equivalent to selecting Holdoff from the Trig menu and then viewing or setting the trigger Level or selecting B Event (Delayed) Trigger Setup from the Trig menu and setting the B Trig Level voltage.

**Group** Trigger

**Syntax** TRIGGER:{A|B}:LEVEL {ECL|TTL|<NR3>}  
 TRIGGER:{A|B}:LEVEL?

**Related Commands** TRIGGER:{A|B}:LEVEL, TRIGGER:B, TRIGGER:{A|B}:EDGE:SOURCE  
 TRIGGER:B:STATE

**Arguments** ECL specifies the ECL high level.  
 TTL specifies the TTL high level.  
 <NR3> specifies the trigger level in user units (usually volts).

**Examples** TRIGGER:A:LEVEL TTL sets the A edge trigger to TTL high level.  
 TRIGGER:A:LEVEL? might return :TRIGGER:A:LEVEL 1.3000E+00, indicating that the A edge trigger is set to 1.3 V.

## TRIGger:{A|B}:LEVel:CH<x>

This command sets or queries the CH<x> trigger level for TRIGGER:LVL SRC PREFERENCE SRCDEPENDENT mode.

The CH<x> range is 1 to 4.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LEVEL:CH<x> {ECL|TTL|<NR3>}  
 TRIGger:{A|B}:LEVEL:CH<x>?

**Arguments** ECL specifies the ECL high level.  
 TTL specifies the TTL high level.  
 <NR3> specifies the trigger level in user units (usually volts).

**Examples** TRIGGER:A:LEVEL:CH1 TTL sets the A trigger level for Channel 1 to the TTL high level.  
 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|B}:LOGIc? (Query Only)

This query-only command returns all of the logic trigger parameters.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc?

**Related Commands** [TRIGger:{A|B}:LOGIc:CLAss](#)

**Examples** TRIGGER:A:LOGIC? might return :TRIGGER:A:LOGIC:CLASS PATTERN; FUNCTION AND; WHEN TRUE; THRESHOLD:CH1 1.4000; CH2 1.4000; CH3 1.4000; CH4 1.4000; :TRIGGER:A:LOGIC:INPUT:CH1



```
HIGH; CH2 X;CH3 X; :TRIGGER:A:LOGIC:PATTERN:INPUT:CH4
X; :TRIGGER:A:LOGIC:PATTERN:WHEN TRUE;
WHEN:LESSLIMIT 5.0000E-9; MORELIMIT
5.0000E-9; :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE
RISE; THRESHOLD 1.4000;SOURCE CH2;
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.4000;SOURCE
CH1; :TRIGGER:A:LOGIC:SETHOLD:HOLDTIME2.0000E-9; SETTIME
3.0000E-9; :TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE;
:TRIGGER:A:LOGIC:STATE:WHEN TRUE.
```

## TRIGger:{A|B}:LOGic:CLAss

This command sets or queries the class of the Logic Trigger. Used in conjunction with the [TRIGger:{A|B}:TYPE](#) command, this command is equivalent to selecting Logic Pattern, Logic State or Setup/Hold Setup from the Trig menu.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:CLASS {PATtern|STATE|SETHold}  
TRIGger:{A|B}:LOGIC:CLASS?

**Related Commands** [TRIGger:{A|B}:TYPE](#), [TRIGger:{A|B}:PULse:CLAss](#)

**Arguments** PATtern sets the instrument to trigger when the specified logical combinations of Channels 1, 2, 3, and 4 are met.

STATE sets the instrument to trigger when the specified conditions of Channels 1, 2, and 3 are met after the Channel 4 (clock) condition is met.

SETHold sets the instrument to trigger on setup and hold violations between a data source and a clock source. Use one channel input as the clock signal and a second channel input as the data input. The clocking and data levels are used to determine if a clock or data transition has occurred.

**Examples** TRIGGER:A:LOGIC:CLASS PATTERN sets the trigger A logic class to PATtern, which causes the instrument to trigger when the specified logical combinations of Channels 1, 2, 3, and 4 are met.

TRIGGER:A:LOGIC:CLASS? might return :TRIGGER:A:LOGIC:CLASS PATTERN, indicating that the A trigger level for channel 2 is set to PATTERN, which causes the instrument to trigger when the specified logical combinations of Channels 1, 2, 3, and 4 are met.

## TRIGger:{A|B}:LOGIc:FUNcTION

This command sets or queries the logical combination of the input channels for the pattern and state logic triggers. This command is equivalent to selecting Event Trigger Setup from the Trig menu, selecting Pattern or State for the Trigger Type, and setting or viewing the Pattern Type.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:FUNcTION {AND|NAND|NOR|OR}  
TRIGger:{A|B}:LOGIc:FUNcTION?

**Related Commands** [TRIGger:{A|B}:LOGIc:PATtern:WHEn](#), [TRIGger:{A|B}:LOGIc:INPut:CH<x>](#), [TRIGger:{A|B}:LOGIc:THReshold:CH<x>](#)

**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|B}:LOGIc:INPut? (Query Only)

This query-only command returns the logic trigger input expected for Channel 1, 2, and 3. Channel 4 is set or queried with the command [TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>](#).

This command is equivalent to selecting Event Trigger Setup and viewing or setting the Input Threshold for the channels.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:INPut?

**Related Commands** [TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>](#)

**Examples** TRIGGER:A:LOGIC:INPUT? might return :TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X, indicating that a logic high is expected on Channel 1 while Channel 2 and Channel 3 are “don’t care.”

## TRIGger:{A|B}:LOGIc:INPut:ALL

This command sets or queries the A or B logic trigger input condition for all the channels. This command is available when the Trigger Type is set to Pattern/State.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:INPut:ALL <QString>  
TRIGger:{A|B}:LOGIC:INPut:ALL?

**Related Commands** [TRIGger:{A|B}:LOGIc:INPut:FORMat](#)

**Arguments** <QString> specifies the bit pattern for all the channels.

**Examples** TRIGGER:A:LOGIC:INPUT:ALL “0xxxx” sets logic high to channel D0 and “don’t care” condition for CH1 through CH4.

TRIGGER:A:LOGIC:INPUT:ALL? might return :TRIGGER:A:LOGIC:INPUT:ALL “1111xxxx”, indicating that a logic high is set on D3 through D0 and a “don’t care” condition is set on CH1 through CH4.

## TRIGger:{A|B}:LOGIc:INPut:CH<x>

This command sets or queries the A or B logical trigger input for the channel specified by x. The value of x ranges from 1 through 3. Note that CH4 cannot be set or queried with this command. For details about setting this channel, see [TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>](#). This command is equivalent to selecting Event Trigger Setup from the Trig menu and then choosing the desired logical input from the Ch<x> drop-down list, which is located in the Input Threshold group box.

**Group** Trigger

**Syntax** TRIGGER:{A|B}:LOGIC:INPUT:CH<x> {HIGH|LOW|X}  
 TRIGGER:{A|B}:LOGIC:INPUT:CH<x>?

**Related Commands** TRIGGER:{A|B}:LOGIC:PATTERN:INPUT:CH<x>

**Arguments** HIGH specifies the logic high.  
 LOW specifies the logic low.  
 X specifies a “don’t care” state.

**Examples** TRIGGER:A:LOGIC:INPUT:CH2 HIGH sets the A logic trigger input to logic HIGH for Channel 2.  
 TRIGGER:A:LOGIC:INPUT:CH1? might return :TRIGGER:LOGIC:INPUT:CH1 X, indicating that the setting for the A logic trigger input to Channel 1 does not matter.

## TRIGGER:{A|B}:LOGIC:INPUT:FORMAT

This command sets or queries the A or B logic trigger pattern format.

**Group** Trigger

**Syntax** TRIGGER:{A|B}:LOGIC:INPUT:FORMAT {HEXadecimal|BINARY}  
 TRIGGER:{A|B}:LOGIC:INPUT:FORMAT?

**Related Commands** TRIGGER:{A|B}:LOGIC:INPUT:ALL

**Arguments** HEXadecimal specifies hexadecimal pattern format for A or B logic trigger.  
 BINARY specifies binary pattern format for A or B logic trigger.

**Examples** TRIGGER:A:LOGIC:INPUT:FORMAT HEX sets the A logic trigger pattern format to hexadecimal.  
 TRIGGER:A:LOGIC:INPUT:FORMAT? might return :TRIGGER:A:LOGIC:INPUT:FORMAT BINARY, indicating that the A logic trigger pattern format is set to binary.

## TRIGger:{A|B}:LOGIc:PATtern? (Query Only)

This query-only command returns the conditions used for generating an A logic pattern trigger, with respect to the defined input pattern, and identifies the maximum and minimum time that the selected pattern may be true and still generate the trigger. This command is equivalent to selecting Logic Pattern from the Trig menu and then viewing the current setups.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:PATtern?

**Related Commands** [TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>](#), [TRIGger:{A|B}:LOGIc:PATtern:WHEn](#), [TRIGger:{A|B}:LOGIc:PATtern:WHEn:LESSLimit](#), [TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit](#)

**Examples** TRIGGER:A:LOGIC:PATTERN? might return  
 :TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 X;  
 :TRIGGER:A:LOGIC:PATTERN:WHEN TRUE; WHEN:LESSLIMIT  
 5.0000E-9;MORELIMIT 5.0000E-9.

## TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>

This command sets or queries the A or B logic trigger input for the specified channel. This command specifies the logic value used when the pattern trigger detects the threshold level.

This command is equivalent to selecting Logic Pattern from the Trig menu and then choosing the desired logical input from the channel drop-down list, which is located in the Input Threshold group box.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:PATtern:INPut:CH<x> {HIGH|LOW|X}  
 TRIGger:{A|B}:LOGIC:PATtern:INPut:CH<x>?

**Related Commands** [TRIGger:{A|B}:LOGIc:FUNcTION](#), [TRIGger:{A|B}:LOGIc:INPut:CH<x>](#), [TRIGger:{A|B}:LOGIc:THREshold:CH<x>](#)

**Arguments** HIGH specifies the logic high.

LOW specifies the logic low.

X specifies a “do not care” state.

- Examples**    `TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH` This command sets the A logic trigger input to logic high for Channel 4 when the logic class is set to `PATtern`. When the threshold level is detected, High places a 1 on the Channel 4 input to the selected function.
- `TRIGGER:A:LOGIC:PATTERN:INPUT:CH4?` might return `:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH`, indicating that the logic input for Channel 4 is logic high.

## TRIGger:{A|B}:LOGIc:PATtern:WHEn

This command sets or queries the condition for generating an A or B logic pattern trigger with respect to the defined input pattern. This command is equivalent to selecting A or B Event (Main) Trigger Setup from the Trig menu, selecting Pattern for Trigger Type, and choosing a trigger condition from the Pattern drop-down list, which is located in the Trigger When group box.

**Group**    Trigger

**Syntax**    `TRIGger:{A|B}:LOGIc:PATtern:WHEn`  
`{TRUE|FALSE|LESSThan|MOREThan}`  
`TRIGger:{A|B}:LOGIc:PATtern:WHEn?`

**Related Commands**    [TRIGger:{A|B}:LOGIc:FUNcTION](#), [TRIGger:{A|B}:LOGIc:INPut?](#), [TRIGger:{A|B}:LOGIc:THReshold?](#), [TRIGger:{A|B}:LOGIc:PATtern:WHEn:LESSLimit](#), [TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit](#)

**Arguments**    `TRUE` sets the instrument to trigger when the pattern becomes true.

`FALSE` sets the instrument to trigger when the pattern becomes false.

`LESSThan` sets the instrument to trigger if the specific pattern is true less than the time set by the [TRIGger:{A|B}:LOGIc:PATtern:WHEn:LESSLimit](#) command.

`MOREThan` argument sets the instrument to trigger if the specific pattern is true longer than the specified time set by the [TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit](#) command.

**Examples**    `TRIGGER:A:LOGIC:PATTERN:WHEN FALSE` specifies to trigger the A logic pattern when the pattern becomes false.

TRIGGER:A:LOGIC:PATTERN:WHEN? might return  
:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE, indicating that the A logic pattern  
will trigger when the pattern becomes true.

## TRIGger:{A|B}:LOGIc:PATtern:WHEn:LESSLimit

This command sets or queries the maximum time that the selected pattern may be true and still generate an A or B logic pattern trigger. This command is equivalent to selecting the A or B Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting Less Than for the Pattern in the Trigger When settings, and entering a maximum value for Time.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:PATtern:WHEn:LESSLimit <NR3>  
TRIGger:{A|B}:LOGIC:PATtern:WHEn:LESSLimit?

**Related Commands** [TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit](#)

**Arguments** <NR3> specifies the amount of time to hold the pattern true.

**Examples** TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT 10.0E+00 sets the maximum  
time that the selected pattern may hold true (and generate an A logic pattern  
trigger) to 10 s.

TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT? might return  
:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT 5.0000E-09, indicating  
that the selected pattern may hold true for up to 5 ns and still generate an A logic  
pattern trigger.

## TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit

This command sets or queries the minimum time that the selected pattern may be true and still generate an A or B logic pattern trigger. This command is equivalent to selecting A or B Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting More Than for the Pattern in the Trigger When settings, and entering a minimum value for Time.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:PATtern:WHEn:MORELimit <NR3>  
 TRIGger:{A|B}:LOGIC:PATtern:WHEn:MORELimit?

**Related Commands** [TRIGger:{A|B}:LOGIC:PATtern:WHEn:LESSLimit](#)

**Arguments** <NR3> specifies the amount of time to hold the pattern true.

**Examples** TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT 10.0E+00 sets the minimum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT? might return :TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT 5.0000E-09, indicating that the selected pattern must hold true for at least 5 ns to generate an A logic pattern trigger.

## TRIGger:{A|B}:LOGIC:SETHold? (Query Only)

This query-only command returns the clock edge polarity, voltage threshold and source input; data voltage threshold and source; and both setup and hold times for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current setups.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:SETHo]d?

**Related Commands** [TRIGger:{A|B}:LOGIC:PATtern?](#), [TRIGger:{A|B}:LOGIC:STATE?](#)

**Examples** TRIGGER:A:LOGIC:SETHOLD? might return the settings :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE; THRESHOLD 1.4000;THRESHOLD:CH1 1.4000;CH2 1.2000;CH3 1.2000;CH4 1.2000;; TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH2;; TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.4000;THRESHOLD: CH1 1.4000;CH2 1.2000;CH3 1.2000;CH4 1.2000;; TRIGGER:A:LOGIC:SETHOLD :HOLDTIME 2.0000E-9;SETTIME 3.0000E-9;QUALIFY OCCURS.



## TRIGger:{A|B}:LOGIc:SETHold:CLOCK? (Query Only)

This query-only command returns the clock edge polarity, voltage threshold and source input for setup and hold triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current clock setups.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:SETHold:CLOCK?

**Related Commands** [TRIGger:{A|B}:LOGIC:SETHold:DATA?](#), [TRIGger:{A|B}:LOGIC:SETHold:CLOCK:EDGE](#), [TRIGger:{A|B}:LOGIC:SETHold:CLOCK:THReshold](#), [TRIGger:{A|B}:LOGIC:SETHold:CLOCK:SOUrce](#)

**Examples** TRIGGER:A:LOGIC:SETHOLD:CLOCK? might return  
 :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD  
 1.2000;THRESHOLD:CH1 1.2000;CH2 1.2000; CH3 1.2000;CH4  
 1.2000;:TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH2 indicating the  
 current clock settings for setup and hold triggering.

## TRIGger:{A|B}:LOGIc:SETHold:CLOCK:EDGE

This command sets or queries the clock edge polarity for setup and hold triggering. This is equivalent to selecting Setup/Hold Setup from the Trig menu and then choosing the desired Clock Edge.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:SETHold:CLOCK:EDGE {FALL|RISe}  
 TRIGger:{A|B}:LOGIC:SETHold:CLOCK:EDGE?

**Related Commands** [TRIGger:{A|B}:LOGIC:SETHold:CLOCK:SOUrce](#), [TRIGger:{A|B}:LOGIC:SETHold:CLOCK:THReshold](#)

**Arguments** FALL specifies polarity as the clock falling edge.  
 RISe specifies polarity as the clock rising edge.

**Examples** TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE specifies the polarity as the  
 clock rising edge.

TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE? might return  
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE, indicating that  
polarity is specified as the clock rising edge.

## TRIGger:{A|B}:LOGIc:SETHold:CLOCK:SOURce

This command sets or queries the clock source for the A or B logic trigger setup and hold input. This is equivalent to selecting Setup/Hold Setup from the Trig menu and choosing the desired channel from the Clock Source drop-down list.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:SOURce CH<x> TRIGger:{A B}:LOGIc:SETHold:CLOCK:SOURce?
<b>Related Commands</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:EDGE, TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold
<b>Arguments</b>	CH<x> specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two channel instruments.
<b>Examples</b>	TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH1 specifies channel 1 as the A logic setup and hold input.  TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE? might return :TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH4, indicating that channel 4 is the clock source for the setup and hold input. For the A logic trigger.

## TRIGger:{A|B}:LOGIc:SETHold:CLOCK:THReshold

This command sets or queries the clock voltage threshold for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and setting the desired Clock Level.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold {ECL TTL <NR3>} TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold?

<b>Related Commands</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:EDGE, TRIGger:{A B}:LOGIc:SETHold:CLOCK:SOUrce
<b>Arguments</b>	ECL specifies a preset ECL high level. TTL specifies a preset TTL high level. <NR3> is the clock level, in volts.
<b>Examples</b>	TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD TTL specifies the preset TTL value as the clock threshold for the setup and hold trigger.  TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD? might return :TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD 1.2000E+00, indicating that the clock threshold for the setup and hold trigger is 1.2 V.

## TRIGger:{A|B}:LOGIc:SETHold:CLOCK:THReshold:CH<x>

This command sets or queries the clock voltage threshold for setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Clock Level. The value of x can range from 1 through 4.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold:CH<x> {ECL TTL <NR3>} TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold:CH<x>?
<b>Related Commands</b>	TRIGger:{A B}:LOGIc:SETHold:CLOCK:EDGE, TRIGger:{A B}:LOGIc:SETHold:CLOCK:SOUrce
<b>Arguments</b>	ECL specifies a preset ECL high level. TTL specifies a preset TTL high level. <NR3> is the clock level, in volts.
<b>Examples</b>	TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD:CH3 TTL specifies the preset TTL value as the clock threshold for setup and hold trigger for Channel 3.  TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD:CH3? might return :TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD:CH3 1.2000E+00, indicating that the clock threshold for setup and hold trigger is 1.2 V for Channel 3.

## TRIGger:{A|B}:LOGIc:SETHold:DATA? (Query Only)

This query-only command returns the voltage threshold and data source for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current data setups.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:SETHoId:DATA?

**Related Commands** [TRIGger:{A|B}:LOGIc:SETHold:CLOCK?](#)

**Examples** TRIGGER:A:LOGIC:SETHOLD:DATA? might return  
 :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD  
 1.2000;THRESHOLD:CH1 1.2000;CH2 1.2000; CH3 1.2000;CH4  
 1.2000;;TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH2, indicating the  
 current trigger data settings.

## TRIGger:{A|B}:LOGIc:SETHold:DATA:SOURce

This command sets or queries the data source for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and choosing the desired channel from the Data Source drop-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:SETHoId:DATA:SOURce CH<x>  
 TRIGger:{A|B}:LOGIc:SETHoId:DATA:SOURce?

**Related Commands** [TRIGger:{A|B}:LOGIc:SETHold:DATA:THReshold](#)

**Arguments** CH<x> specifies the input channel, which ranges from 1 through 4.

**Examples** TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH1 sets channel 1 as the clock  
 source for the setup and hold trigger.

TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE? might return  
 :TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH2 indicating that Channel 2  
 is the current clock source for the setup and hold trigger.

## TRIGger:{A|B}:LOGIc:SETHold:DATA:THReshold

This command sets or queries the data voltage threshold for setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Data Level.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:SETHold:DATA:THReshold {ECL|TTL|<NR3>}  
TRIGger:{A|B}:LOGIC:SETHold:DATA:THReshold?

**Related Commands** [TRIGger:{A|B}:LOGIc:SETHold:DATA:SOUrce](#)

**Arguments** ECL specifies the preset ECL high level.  
TTL specifies the preset TTL high level.  
<NR3> is the setup and hold data level, in V.

**Examples** TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD TTL specifies the preset TTL high level as the current data voltage level for the setup and hold trigger.  
TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD? might return  
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.2000E+00, indicating that 1.2 V is the current data voltage level for the setup and hold trigger.

## TRIGger:{A|B}:LOGIc:SETHold:DATA:THReshold:CH<x>

This command sets or queries the data voltage threshold for setup and hold trigger. This command is equivalent to selecting A or B Event Trigger Setup from the Trig menu and then setting the desired Data Level. The value of x can range from 1 through 4.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:SETHold:DATA:THReshold:CH<x>  
{ECL|TTL|<NR3>}  
TRIGger:{A|B}:LOGIC:SETHold:DATA:THReshold:CH<x>?

**Related Commands** [TRIGger:{A|B}:LOGIc:SETHold:DATA:THReshold:CH<x>](#)

<b>Arguments</b>	ECL specifies the preset ECL high level. TTL specifies the preset TTL high level. <NR3> is the setup and hold data level, in V.
<b>Examples</b>	TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD:CH2 TTL specifies the preset TTL high level as the current data voltage level for setup and hold trigger for Channel 2.  TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD:CH2? might return :TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD:CH2 1.2000E+00, indicating that 1.2 V is the current data voltage level for setup and hold trigger for Channel 2.

## TRIGger:{A|B}:LOGIc:SETHold:HOLDTime

This command sets or queries 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.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIc:SETHold:HOLDTime <NR3> TRIGger:{A B}:LOGIc:SETHold:HOLDTime?
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:LOGIc:SETHold:SETTime</a>
<b>Arguments</b>	<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.
<b>Examples</b>	TRIGGER:A:LOGIC:SETHOLD:HOLDTIME 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.  TRIGGER:A:LOGIC:SETHOLD:HOLDTIME? might return :TRIGGER:A:LOGIC:SETHOLD:HOLDTIME 2.0000E-09, indicating that the current hold time for the setup and hold trigger is 2 ns.

## TRIGger:{A|B}:LOGIc:SETHold:QUALify

This command sets or queries the Setup/Hold Trigger qualification. This is equivalent to selecting Setup/Hold Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Setup/Hold drop-down list box.

---

**NOTE.** If you use the *LOGIC* argument, you must specify the channel logic conditions.

*BUS* is same as *OccursAnd* in the user interface.

---

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:{A B}:LOGIC:SETHold:QUALify {OCCurs LOGIC BUS} TRIGger:{A B}:LOGIC:SETHold:QUALify?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:QUALification:BUS:FORMat</a> , <a href="#">TRIGger:QUALification:BUS:SOUrce</a> , <a href="#">TRIGger:QUALification:BUS:VALue</a>
<b>Arguments</b>	<p><b>OCCurs</b> specifies a trigger if any detectable event occurs.</p> <p><b>LOGIC</b> specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the <a href="#">TRIGger:{A B}:LOGIC:INPut:CH&lt;x&gt;</a> and <a href="#">TRIGger:{A B}:LOGIC:THReshold:CH&lt;x&gt;</a> commands.</p> <p><b>BUS</b> specifies a trigger if the bus input qualifications meet the pattern set by the <a href="#">TRIGger:QUALification:BUS:VALue</a> command.</p>
<b>Examples</b>	<p><code>TRIGGER:A:LOGIC:SETHOLD:QUALIFY OCCURS</code> sets the trigger qualifier to <code>OCCurs</code>, which initiates a trigger if the instrument detects any setup and hold trigger qualifications.</p> <p><code>TRIGGER:A:LOGIC:SETHOLD:QUALIFY?</code> might return  <code>:TRIGGER:A:LOGIC:SETHOLD:QUALIFY OCCURS</code>, indicating that a trigger will occur if the instrument detects any of the setup and hold qualifications.</p>

## TRIGger:{A|B}:LOGIC:SETHold:SETTime

This command sets or queries 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.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:{A B}:LOGIC:SETHold:SETTime &lt;NR3&gt; TRIGger:{A B}:LOGIC:SETHold:SETTime?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:LOGIC:SETHold:HOLDTime</a>

**Arguments** <NR3> specifies the setup time for setup and hold violation triggering.

**Examples** TRIGGER:A:LOGIC:SETHOLD:SETTIME 3.0E-6 specifies that the current setup time for setup and hold trigger is 3  $\mu$ s.

TRIGGER:A:LOGIC: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|B}:LOGIc:STATE? (Query Only)

This query-only command returns the data input and trigger criteria for the logic trigger. This command is equivalent to selecting Logic State from the Trig menu and then viewing the current logic state settings.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:STATE?

**Related Commands** [TRIGger:{A|B}:LOGIc:STATE:INPut:CH<x>](#), [TRIGger:{A|B}:LOGIc:STATE:WHEn](#)

**Examples** TRIGGER:A:LOGIC:STATE? might return  
:TRIGGER:A:LOGIC:STATE:INPUT:CH4  
RISE;;TRIGGER:A:LOGIC:STATE:WHEN TRUE;CLOCK:SOURCE CH4.

## TRIGger:{A|B}:LOGIc:STATE:CLOck:SOUrce

This command sets or queries the clock source for the A or B logic trigger.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:STATE:CLOck:SOUrce {CH4|CQ}  
TRIGger:{A|B}:LOGIc:STATE:CLOck:SOUrce?

---

**NOTE.** CQ is available only on MSO70000 series of products.

---



**Arguments** CH4 specifies the analog channel as clock source.  
 positive  
 CQ specifies the channel/qualifier as clock source.

**Examples** TRIGGER:A:LOGIC:STATE:CLOCK:SOURCE CQ sets the clock source to channel/qualifier.  
 TRIGGER:A:LOGIC:STATE:CLOCK:SOURCE? might return  
 :TRIGGER:A:LOGIC:STATE:CLOCK:SOURCE CH4, indicating that the clock source is set to CH4.

## TRIGger:{A|B}:LOGIc:STATE:INPut:CH<x>

This command sets or queries the slope for the channel specified by x when the logic class is set to STATE. This command is equivalent to selecting Logic State from the Trig menu and then choosing the desired channel input (NEG or POS) from the Ch drop-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIc:STATE:INPut:CH<x> {FALL|RISe}  
 TRIGger:{A|B}:LOGIc:STATE:INPut:CH<x>?

**Related Commands** [TRIGger:{A|B}:LOGIc:STATE:WHEn](#)

**Arguments** FALL specifies the falling edge.  
 RISe specifies the rising edge.

**Examples** TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISe specifies that the A logic trigger input for Channel 4 is the rising edge.  
 TRIGGER:A:LOGIC:STATE:INPUT:CH4? might return  
 :TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISe, indicating that the A logic trigger input for Channel 4 is the rising edge.

## TRIGger:{A|B}:LOGIc:STATE:WHEn

This command sets or queries the condition for generating an A or B logic state trigger. This command is equivalent to selecting Logic State from the Trig menu and choosing the desired condition from the Trigger When Pattern drop-down list.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIC:STATE:WHEN {TRUE FALSE} TRIGger:{A B}:LOGIC:STATE:WHEN?
<b>Related Commands</b>	TRIGger:{A B}:LOGIC:INPut:CH<x>, TRIGger:{A B}:LOGIC:STATE:INPut:CH<x>
<b>Arguments</b>	TRUE specifies that the trigger occurs when the clock transition on channel 4 occurs and the pattern of channels 1-3 are at the desired logic input states.  FALSE specifies that the trigger occurs when the desired clock transition on channel 4 occurs and the desired logic input states on channels 1-3 are not found.
<b>Examples</b>	TRIGGER:A:LOGIC:STATE:WHEN TRUE specifies that the logic condition to trigger upon is true.  TRIGGER:A:LOGIC:STATE:WHEN? might return :TRIGGER:A:LOGIC:STATE:WHEN FALSE, indicating that the logic condition to trigger upon is false.

## TRIGger:{A|B}:LOGIC:THReshold? (Query Only)

This query-only command returns the threshold voltage for all channels in a logic trigger. This command query is equivalent to selecting Event Trigger Setup from the Trig menu, choosing a logic trigger type, such as State or Pattern, and viewing the current Input Threshold voltage settings.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:LOGIC:THReshoId?
<b>Related Commands</b>	TRIGger:{A B}:LOGIC:THReshold:CH<x>, TRIGger:{A B}:LOGIC:INPut?, TRIGger:{A B}:LOGIC:INPut:CH<x>
<b>Examples</b>	TRIGGER:A:LOGIC:THRESHOLD? might return :TRIGGER:A:LOGIC:THRESHOLD CH1 24.0000E-03; CH2 1.2000E+00; CH3 1.2000E+00; CH4 1.2000E+00, indicating the threshold voltages for the channels in an A logic trigger are as follows: channel 1 = 24 mV; channel 2 = 1.2 V; channel 3 = 1.2 V; channel 4 = 1.2 V.

**TRIGger:{A|B}:LOGIc:THReshold:CH<x>**

This command sets or queries the A or B logic trigger threshold voltage for the channel, specified by x, which ranges from 1 through 4. This command is equivalent to selecting A or B Event Trigger Setup from the Trig menu, choosing a logic trigger type, such as State or Pattern, and setting the Input Threshold voltage for the desired channel.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOGIC:THReshold:CH<x> <NR3>  
TRIGger:{A|B}:LOGIC:THReshold:CH<x>?

**Related Commands** [TRIGger:{A|B}:LOGIc:INPut?](#)

**Arguments** <NR3> specifies the threshold voltage.

**Examples** TRIGGER:A:LOGICTHRESHOLD:CH2 3.0E-3 sets the A logic trigger threshold voltage for Channel 2 to 3 mV.

TRIGGER:A:LOGIC:THRESHOLD:CH3? might return  
:TRIGGER:A:LOGIC:THRESHOLD:CH3 1.2000E+00, indicating that the A logic trigger threshold voltage for Channel 3 is 1.2 V.

**TRIGger:{A|B}:LOWerthreshold:CH<x>**

This command sets or queries the A or B lower trigger level for TRIGger:LVLsrcpreference[TRIGger:LVLsrcpreference](#) SRCDependent or SRCIndependent modes for the channel, specified by x, which ranges from 1 through 4.

**Group** Trigger

**Syntax** TRIGger:{A|B}:LOWerthreshold:CH<x> {ECL|TTL|<NR3>}  
TRIGger:{A|B}:LOWerthreshold:CH<x>?

**Related Commands** [TRIGger:LVLsrcpreference](#), [TRIGger:{A|B}:UPPerthreshold:CH<x>](#)

**Arguments** ECL specifies the preset ECL high level.

TTL specifies the preset TTL high level.

<NR3> specifies the threshold voltage in user units.

**Examples** TRIGGER:A:LOWERTHRESHOLD:CH2 TTL sets the A trigger threshold voltage for Channel 2 to the TTL low level.

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|B}:PULse? (Query Only)

This query-only command returns the pulse trigger parameters.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse?

**Related Commands** [TRIGger:{A|B}:EDGE?](#), [TRIGger:{A|B}:LOGic?](#)

**Examples** TRIGGER:A:PULSE? might return :TRIGGER:A:PULSE:CLASS GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;POLARITY POSITIVE; POLARITY:CH1 POSITIVE;CH2 POSITIVE;CH3 POSITIVE;CH4 POSITIVE;; TRIGGER:A:PULSE:GLITCH:QUALIFY OCCURS;; TRIGGER:A:PULSE:WINDOW:TYPE INSIDE;WHEN OCCURS;WIDTH 2.0000E-9;; TRIGGER:A:PULSE:TRANSITION:DELTATIME 2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH 1.2000;LOW 800.0000E-3;; TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN; :TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;POLARITY POSITIVE;; TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYSHIGH;TIME 2.0000E-9.

## TRIGger:{A|B}:PULse:CLAss

This command sets or queries the type of pulse on which to trigger. This command is equivalent to selecting the setup menu for the pulse type that you want from the Trig menu: Glitch Setup, Width Setup, Runt Setup, Timeout Setup, or Transition Setup.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:CLASS {GLItch|RUNT|WIDTh|  
TRANSition|TIMEOut|WINDow}  
TRIGger:{A|B}:PULse:CLASS?

**Related Commands** TRIGger:{A|B}:PULse:GLItch?, TRIGger:{A|B}:PULse:RUNT?,  
TRIGger:{A|B}:PULse:WIDTh?, TRIGger:{A|B}:PULse:TIMEOut?,  
TRIGger:{A|B}:PULse:TRANSition?, TRIGger:{A|B}:TYPE

**Arguments** GLItch triggers when a pulse is found that is of the specified polarity and width. These are set with the commands TRIGger:{A|B}:PULse:GLItch:POLarity and TRIGger:{A|B}:PULse:GLItch:WIDTh.

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 TRIGger:{A|B}:PULse:RUNT:THReshold:LOW and TRIGger:{A|B}:PULse:RUNT:THReshold:HIGh commands.

WIDTh triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by TRIGger:{A|B}:PULse:WIDTh:LOWLimit and TRIGger:{A|B}:PULse:WIDTh:HIGHLimit. The polarity is selected using the TRIGger:{A|B}:PULse:WIDTh:POLarity command.

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.

TIMEOut triggers when the pulse train stops in the selected state for longer than the specified time.

WINDow triggers when a pulse is found that meets the conditions set by the A Event window trigger type, specified by the following commands: TRIGger:{A|B}:PULse:WINDow:THReshold:HIGh, TRIGger:{A|B}:PULse:WINDow:THReshold:LOW, TRIGger:{A|B}:PULse:WINDow:THReshold:BOTh, and TRIGger:{A|B}:PULse:WINDow:WIDTh

**Examples** TRIGGER:A:PULSE:CLASS WIDTH specifies a width pulse for the A trigger.

TRIGGER:A:PULSE:CLASS? might return :TRIGGER:A:PULSE:CLASS GLITCH, indicating that a pulse was found that is of the specified polarity and width.

## TRIGger:{A|B}:PULse:GLItch? (Query Only)

This query-only command returns the current glitch pulse trigger parameters. This command query is equivalent to selecting Glitch Setup from the Trig menu and viewing the current glitch trigger settings.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:GLItch?
<b>Related Commands</b>	TRIGger:{A B}:PULse:GLItch:POLarity, TRIGger:{A B}:PULse:GLItch:TRIGIF, TRIGger:{A B}:PULse:GLItch:WIDTH
<b>Examples</b>	TRIGGER:A:PULSE:GLITCH? might return :TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09; FILTER ACCEPT;POLARITY POSITIVE.

## TRIGger:{A|B}:PULse:GLItch:POLarity

This command sets or queries the polarity for A or B pulse glitch trigger. This command is equivalent to selecting Glitch Setup from the Trig menu and then choosing the desired Polarity.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:GLItch:POLarity {POSITIVE NEGAtive EITHer} TRIGger:{A B}:PULse:GLItch:POLarity?
<b>Related Commands</b>	TRIGger:{A B}:LEVel, TRIGger:{A B}:PULse:GLItch:WIDTH
<b>Arguments</b>	<p>POSITIVE specifies that the instrument will only trigger when the polarity of the glitch is positive.</p> <p>NEGAtive specifies that the instrument will only trigger when the polarity of the glitch is negative.</p> <p>EITHer specifies that the instrument will trigger when the polarity of the glitch is either positive or negative.</p>
<b>Examples</b>	<p>TRIGGER:A:PULSE:GLITCH:POLARITY EITHER specifies that the polarity of the glitch can be either positive or negative for the A trigger to occur.</p> <p>TRIGGER:A:PULSE:GLITCH:POLARITY? might return :TRIGGER:A:PULSE:GLITCH:POLARITY POSITIVE, indicating that the polarity of the glitch must be positive for the trigger to occur.</p>

## TRIGger:{A|B}:PULse:GLItch:POLarity:CH<x>

This command sets or queries the polarity for A or B pulse glitch trigger for the channel.

This command is equivalent to selecting Glitch Setup from the Trig menu and then choosing the desired Polarity.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:GLItch:POLarity:CH<x>  
{EITHer|NEGAtive|POSITIVe}  
TRIGger:{A|B}:PULse:GLItch:POLarity:CH<x>?

**Related Commands** TRIGger:{A|B}:LEVel, TRIGger:{A|B}:PULse:GLItch:WIDth

**Arguments** EITHer specifies that the instrument will trigger when the polarity of the glitch is either positive or negative.

NEGAtive specifies that the instrument will only trigger when the polarity of the glitch is negative.

POSITIVe specifies that the instrument will only trigger when the polarity of the glitch is positive.

**Examples** TRIGGER:A:PULSE:GLITCH:POLARITY:CH2 EITHer specifies that the polarity of the glitch can be either positive or negative for the Channel 2 trigger to occur.

TRIGGER:A:PULSE:GLITCH:POLARITY:CH2? might return  
:TRIGGER:A:PULSE:GLITCH:POLARITY:CH2 POSITIVE, indicating that the polarity of the glitch must be positive for the Channel 2 trigger to occur.

## TRIGger:{A|B}:PULse:GLItch:QUALify

This command sets or queries the Glitch Trigger qualification. This is equivalent to selecting Glitch Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Glitch drop-down list box.

---

**NOTE.** *If you use the LOGIC argument, you must specify the channel logic conditions.*

*BUS is same as OccursAnd in the user interface.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:GLItch:QUAlify {OCCurs LOGIC BUS} TRIGger:{A B}:PULse:GLItch:QUAlify?
<b>Related Commands</b>	TRIGger:QUALification:BUS:FORMat, TRIGger:QUALification:BUS:SOUrce, TRIGger:QUALification:BUS:VALue
<b>Arguments</b>	OCCurs specifies a trigger if any detectable event occurs.  LOGIC specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the TRIGger:{A B}:LOGIc:INPut:CH<x> and TRIGger:{A B}:LOGIc:THReshold:CH<x> commands.  BUS specifies a trigger if the bus input qualifications meet the pattern set by the TRIGger:QUALification:BUS:VALue command.
<b>Examples</b>	TRIGGER:A:PULSE:GLITCH:QUALIFY OCCURS This command sets the trigger qualifier to OCCURS, which initiates a trigger if the instrument detects any glitch trigger qualifications.  TRIGGER:A:PULSE:GLITCH:QUALIFY?, indicating that a trigger will occur if the instrument detects any of the glitch qualifications.

## TRIGger:{A|B}:PULse:GLItch:TRIGIF

This command sets or queries the acceptance/rejection of the glitch pulse trigger, based on width. This command is equivalent to selecting Glitch Setup from the Trig menu and choosing the desired Trig if Width setting.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:GLItch:TRIGIF {ACCEpt REJect} TRIGger:{A B}:PULse:GLItch:TRIGIF?
<b>Related Commands</b>	TRIGger:{A B}:PULse:GLItch:WIDth
<b>Arguments</b>	ACCEpt specifies that the instrument will only trigger on pulses that are narrower than the specified width, when the trigger type is set to glitch. The width is specified using the TRIGger:{A B}:PULse:GLItch:WIDth command.



REJECT specifies that the instrument will only trigger on pulses that are wider than the specified width, when the trigger type is set to glitch. The width is specified using the `TRIGGER:{A|B}:PULSE:GLITCH:WIDTH` command.

**Examples** `TRIGGER:A:PULSE:GLITCH:TRIGIF REJECT` specifies that the instrument triggers on pulses that are wider than the specified width.

`TRIGGER:A:PULSE:GLITCH:TRIGIF ACCEPT?` might return `:TRIGGER:A:PULSE:GLITCH:TRIGIF ACCEPT`, indicating that the instrument is set to trigger on pulses that are narrower than the specified width.

## TRIGGER:{A|B}:PULSE:GLITCH:WIDTH

This command sets or queries the width for the glitch trigger. This command is equivalent to selecting Glitch Setup from the Trig menu and then setting the desired Width.

For information about using the width value, refer to the `TRIGGER:{A|B}:PULSE:GLITCH:TRIGIF` command.

**Group** Trigger

**Syntax** `TRIGGER:{A|B}:PULSE:GLITCH:WIDTH <NR3>`  
`TRIGGER:{A|B}:PULSE:GLITCH:WIDTH?`

**Related Commands** `TRIGGER:{A|B}:LEVEL`, `TRIGGER:{A|B}:PULSE:GLITCH:TRIGIF`

**Arguments** `<NR3>` argument specifies the width of the glitch in seconds.

**Examples** `TRIGGER:A:PULSE:GLITCH:WIDTH 15E-6` sets the width of the glitch to 15  $\mu$ s.

`TRIGGER:A:PULSE:GLITCH:WIDTH?` might return `:TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09`, indicating that the width of the glitch is currently set at 2 ns.

## TRIGGER:{A|B}:PULSE:RUNT? (Query Only)

This query-only command returns the current runt trigger parameters. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT?

**Related Commands** [TRIGger:{A|B}:PULse:GLItch?](#), [TRIGger:{A|B}:PULse:TIMEOut?](#),  
[TRIGger:{A|B}:PULse:TRANSition?](#), [TRIGger:{A|B}:PULse:WIDth?](#)

**Examples** TRIGGER:A:PULSE:RUNT? might return  
:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE;POLARITY:CH1  
POSITIVE;CH2 POSITIVE;CH3 POSITIVE;CH4  
POSITIVE;;TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGH 1.2000;LOW  
800.0000E-3;;TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH  
500.0000E-12;QUALIFY OCCURS.

## TRIGger:{A|B}:PULse:RUNT:POLarity

This command sets or queries the polarity for the A or B pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then choosing the Polarity setting.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT:POLarity {POSITIVE|NEGATIVE|EITHER}  
TRIGger:{A|B}:PULse:RUNT:POLarity?

**Related Commands** [TRIGger:{A|B}:PULse:RUNT:THReshold?](#)

**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 indicates either negative or positive polarity.

**Examples** TRIGGER:A:PULSE:RUNT:POLARITY NEGATIVE sets the polarity of the A pulse runt trigger to negative.  
  
TRIGGER:A:PULSE:RUNT:POLARITY? might return  
:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE, indicating that the polarity of the A pulse runt trigger is set to positive.

**TRIGger:{A|B}:PULse:RUNT:POLarity:CH<x>**

This command sets or queries the polarity for the A or B pulse runt trigger for the channel.

This command is equivalent to selecting Runt Setup from the Trig menu and then choosing the Polarity setting for the channel.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT:POLarity:CH<x>  
{EITHer|NEGATive|POSITive}  
TRIGger:{A|B}:PULse:RUNT:POLarity:CH<x>?

**Related Commands** [TRIGger:{A|B}:PULse:RUNT:THReshold?](#)

**Arguments** EITHer indicates either negative or positive polarity.

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.

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.

**Examples** TRIGGER:A:PULSE:RUNT:POLARITY:CH3 NEGATIVE sets the polarity of the Channel 3 A pulse runt trigger to negative.

TRIGGER:A:PULSE:RUNT:POLARITY:CH3? might return  
:TRIGGER:A:PULSE:RUNT:POLARITY:CH3 POSITIVE, indicating that the polarity of the A pulse runt trigger is set to positive.

**TRIGger:{A|B}:PULse:RUNT:QUALify**

This command sets or queries the Runt Trigger qualification. This is equivalent to selecting Runt Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Runt drop-down list box.

---

**NOTE.** *If you use the LOGic argument, you must specify the channel logic conditions.*

---

*BUS is same as OccursAnd in the user interface.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:RUNT:QUALify {OCCurs LOGic BUS} TRIGger:{A B}:PULse:RUNT:QUALify?
<b>Related Commands</b>	<a href="#">TRIGger:QUALification:BUS:FORMat</a> , <a href="#">TRIGger:QUALification:BUS:SOUrce</a> , <a href="#">TRIGger:QUALification:BUS:VALue</a>
<b>Arguments</b>	OCCurs specifies a trigger if any detectable event occurs.  LOGic specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the command.  BUS specifies a trigger if the bus input qualifications meet the pattern set by the <a href="#">TRIGger:QUALification:BUS:VALue</a> command.
<b>Examples</b>	TRIGGER:A:PULSE:RUNT:QUALIFY OCCURS sets the A trigger to trigger if a qualifying runt occurs.  TRIGGER:A:PULSE:RUNT:QUALIFY? might return :TRIGGER:A:PULSE:RUNT:QUALIFY OCCURS, indicating that a trigger will occur if the instrument detects any of the runt qualifications.

## TRIGger:{A|B}:PULse:RUNT:THReshold? (Query Only)

This query-only command returns the upper and lower thresholds for the pulse runt trigger. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the Upper Level and Lower Level settings.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:RUNT:THReshold?
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:PULse:RUNT:POLarity</a> , <a href="#">TRIGger:{A B}:PULse:RUNT:THReshold:BOTh</a> , <a href="#">TRIGger:{A B}:PULse:RUNT:THReshold:HIGh</a> , <a href="#">TRIGger:{A B}:PULse:RUNT:THReshold:LOW</a>
<b>Examples</b>	TRIGGER:A:PULSE:RUNT:THRESHOLD? might return :TRIGGER:A:PULSE:THRESHOLD:HIGh 2.0000E+00; LOW 8.0000E-01, indicating that the upper threshold is 2 V and that the lower threshold is set to 0.8 V.

## TRIGger:{A|B}:PULse:RUNT:THReshold:BOTH (No Query Form)

This command (no query form) sets the upper and lower switching thresholds for the pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Upper Level and Lower Level voltages.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:RUNT:THReshold:BOTH {TTL ECL}
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:PULse:RUNT:THReshold?</a>
<b>Arguments</b>	TTL sets the upper and lower threshold to the nominal TTL voltage levels. ECL sets the upper and lower threshold to the nominal ECL voltage levels.
<b>Examples</b>	TRIGGER:A:PULSE:RUNT:THRESHOLD:BOTH TTL sets the threshold of the pulse runt trigger to the nominal TTL voltage levels.

## TRIGger:{A|B}:PULse:RUNT:THReshold:HIGh

This command sets or queries the upper limit for the pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and setting the runt trigger Upper Level voltage.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:RUNT:THReshold:HIGh <NR3> TRIGger:{A B}:PULse:RUNT:THReshold:HIGh?
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:PULse:RUNT:THReshold?</a>
<b>Arguments</b>	<NR3> specifies the threshold value, in volts.
<b>Examples</b>	TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGh 120E-3 sets the upper limit of the pulse runt trigger to 120 mV.  TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGh? might return :TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGh 1.1000E+00, indicating that the upper limit of the pulse runt trigger is currently set to 1.1 V.

## TRIGger:{A|B}:PULse:RUNT:THReshold:LOW

This command sets or queries the lower limit for the pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Lower Level voltage.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT:THResho1d:LOW <NR3>  
TRIGger:{A|B}:PULse:RUNT:THResho1d:LOW?

**Related Commands** [TRIGger:{A|B}:PULse:RUNT:THReshold?](#)

**Arguments** <NR3> specifies the threshold value in volts.

**Examples** TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW 50E-3 sets the lower limit of the pulse runt trigger to 50 mV.

TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW? might return  
:TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW 1.2000E-01,  
indicating that the lower limit of the pulse runt trigger is set to 120 mV.

## TRIGger:{A|B}:PULse:RUNT:WHEn

This command sets or queries the type of pulse width the trigger checks for when it detects a runt. This is equivalent to selecting Runt Setup from the Trig menu and choosing the desired Trigger When setting from the drop-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT:WHEn {OCCurs|WIDERthan}  
TRIGger:{A|B}:PULse:RUNT:WHEn?

**Related Commands** [TRIGger:{A|B}:PULse:RUNT:WIDth](#)

**Arguments** OCCurs argument specifies a trigger event if a runt of any detectable width occurs.  
WIDERthan specifies a trigger event if a runt greater than the specified width occurs.

**Examples** TRIGGER:A:PULSE:RUNT:WHEN WIDERTHAN sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

TRIGGER:A:PULSE: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|B}:PULse:RUNT:WIDth

This command sets or queries the minimum width for an Pulse Runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Width.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:RUNT:WIDth <NR3>  
TRIGger:{A|B}:PULse:RUNT:WIDth?

**Related Commands** [TRIGger:{A|B}:PULse:RUNT:WHEn](#)

**Arguments** <NR3> specifies the minimum width, in seconds.

**Examples** TRIGGER:A:PULSE:RUNT:WIDTH 15E-6 sets the minimum width of the pulse runt trigger to 15  $\mu$ s.

TRIGGER:A:PULSE: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|B}:PULse:SOURce

This command sets or queries the source for the pulse trigger. This source parameter applies to all classes of pulse triggers. This command is equivalent to selecting Event Trigger Setup from the Trig menu, selecting the pulse type (Glitch, Width, Runt, Timeout, or Transition), and then choosing the desired channel from the Source pull-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:SOURce {CH<x>|D<x>}  
TRIGger:{A|B}:PULse:SOURce?

<b>Related Commands</b>	<a href="#">TRIGger:{A B}:EDGE:SOURce</a>
<b>Arguments</b>	CH<x> specifies one of the input channels, which range from 1 through 4. D<x> specifies one of the digital inputs, which range from 0 through 15.
<b>Examples</b>	TRIGGER:A:PULSE:SOURCE CH4 sets Channel 4 as the source for the A pulse trigger. TRIGGER:A:PULSE:SOURCE? might return :TRIGGER:A:PULSE:SOURCE CH2, indicating that the Channel 2 is the source for the A pulse trigger.

## TRIGger:{A|B}:PULse:TIMEOut? (Query Only)

This query-only command returns the polarity and time-out duration for the pulse timeout trigger. This command is equivalent to selecting Timeout Setup from the Trig menu and viewing the polarity in the Trigger When box and the Timer setting.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:TIMEOut?
<b>Related Commands</b>	<a href="#">TRIGger:{A B}:PULse:TIMEOut:POLarity</a> , <a href="#">TRIGger:{A B}:PULse:TIMEOut:TIME</a>
<b>Examples</b>	TRIGGER:A:PULSE:TIMEOUT? might return :TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYSHIGH;TIME 2.0000E-9, indicating that the polarity setting for the A pulse trigger is STAYSHIGH (positive) and the timeout duration is 2 ns.

## TRIGger:{A|B}:PULse:TIMEOut:POLarity

This command sets or queries the polarity for the pulse timeout trigger. This command is equivalent to selecting Timeout Setup from the Trig menu and setting the desired polarity in the Trigger When box.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:TIMEOut:POLarity {STAYSHigh STAYSLow EITHer}



TRIGger:{A|B}:PULse:TIMEOut:POLArity?

**Related Commands** [TRIGger:{A|B}:PULse:TIMEOut?](#)

**Arguments** STAYSHigh indicates that a pulse edge must stay high (positive) for the required time period to permit timeout triggering to occur. This is the default polarity.

STAYSLow indicates that a pulse edge must stay low (negative) for the required time period to permit timeout triggering to occur.

EITHer indicates that the polarity of the timeout trigger can stay either high or low (positive or negative) for the required time period to permit time out triggering to occur.

**Examples** TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYSHIGH sets the polarity of the A timeout trigger to positive.

TRIGGER:A:PULSE:TIMEOUT:POLARITY? might return  
:TRIGGER:A:PULSE:TIMEOUT:POLARITY EITHER, indicating that the polarity of the A timeout trigger can be either positive or negative.

## TRIGger:{A|B}:PULse:TIMEOut:POLArity:CH<x>

This command sets or queries the polarity for the A or B pulse timeout trigger for the channel.

This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Polarity for the channel.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TIMEOut:POLArity:CH<x>  
{STAYSHigh|STAYSLow|EITHer}  
TRIGger:{A|B}:PULse:TIMEOut:POLArity:CH<x>?

**Arguments** STAYSHigh indicates that a pulse edge must stay high (positive) for the required time period to permit time out triggering to occur. This is the default polarity.

STAYSLow indicates that a pulse edge must stay low (negative) for the required time period to permit time out triggering to occur.

EITHer indicates that the polarity of the time out trigger can stay either high or low (positive or negative) for the required time period to permit time out triggering to occur.

**Examples** TRIGGER:A:PULSE:TIMEOUT:POLARITY:CH1 STAYSHIGH sets the polarity of the A pulse time out trigger to positive for Channel 1.

TRIGGER:A:PULSE:TIMEOUT:POLARITY:CH1? might return  
:TRIGGER:A:PULSE:TIMEOUT:POLARITY:CH1 EITHER, indicating that the polarity of the A pulse time out trigger for Channel 1 can be either positive or negative.

## TRIGger:{A|B}:PULse:TIMEOut:QUALify

This command sets or queries the Timeout Trigger qualification.

This is equivalent to selecting Timeout Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Timeout drop-down list box.

---

**NOTE.** If you use the LOGIC argument, you must specify the channel logic conditions.

BUS is same as OccursAnd in the user interface.

---

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TIMEOut:QUALify {OCCurs|LOGIC|BUS}  
TRIGger:{A|B}:PULse:TIMEOut:QUALify?

**Related Commands** [TRIGger:QUALification:BUS:FORMat](#), [TRIGger:QUALification:BUS:SOUrce](#), [TRIGger:QUALification:BUS:VALue](#)

**Arguments** OCCurs specifies a trigger if any detectable event occurs.

LOGIC specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the [TRIGger:{A|B}:LOGIC:INPut:CH<x>](#) and [TRIGger:{A|B}:LOGIC:THReshold:CH<x>](#) commands.

BUS specifies a trigger if the bus input qualifications meet the pattern set by the [TRIGger:QUALification:BUS:VALue](#) command.

**Examples** TRIGGER:A:PULSE:TIMEOUT:QUALIFY OCCURS sets the trigger qualifier to OCCURS, which initiates a trigger if the instrument detects any timeout trigger qualifications.

TRIGGER:A:PULSE:TIMEOUT:QUALIFY? might return  
:TRIGGER:A:PULSE:TIMEOUT:QUALIFY OCCURS, indicating that a trigger will occur if the instrument detects any of the timeout qualifications.

## TRIGger:{A|B}:PULse:TIMEOut:TIME

This command sets or queries the pulse timeout trigger time (measured in seconds). This command is equivalent to selecting Timeout Setup from the Trig menu and setting a value for Timer.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TIMEOut:TIME <NR3>  
TRIGger:{A|B}:PULse:TIMEOut:TIME?

**Related Commands** [TRIGger:{A|B}:PULse:TIMEOut?](#)

**Arguments** <NR3> argument specifies the timeout period in seconds.

**Examples** TRIGGER:A:PULSE:TIMEOUT:TIME 3.134E-6 sets the timeout to 3.134  $\mu$ s.  
TRIGGER:A:;PULSE:TIMEOUT:TIME? might return  
:TRIGGER:A:PULSE:TIMEOUT:TIME 2.0000E-9, indicating that the timeout is currently set to 2 ns.

## TRIGger:{A|B}:PULse:TRANSition? (Query Only)

This query-only command returns delta time, polarity, and both upper and lower threshold limits for the transition time trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then viewing the current transition settings.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition?

**Related Commands** [TRIGger:{A|B}:TYPE](#), [TRIGger:{A|B}:PULse:CLAss](#), [TRIGger:{A|B}:PULse:GLItch?](#), [TRIGger:{A|B}:PULse:RUNT?](#), [TRIGger:{A|B}:PULse:TIMEOut?](#), [TRIGger:{A|B}:PULse:WIDth?](#)

**Examples** TRIGGER:A:PULSE:TRANSITION? might return  
:TRIGGER:A:PULSE:TRANSITION:DELTATIME  
2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH 1.2000;LOW

800.0000E-3; :TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN  
 indicating the current transition time trigger settings.

## TRIGger:{A|B}:PULse:TRANSition:DELTATime

This command sets or queries the delta time used in calculating the transition value for the transition trigger. This is equivalent to selecting Transition Setup from the Trig menu and setting the Time.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition:DELTATime <NR3>  
 TRIGger:{A|B}:PULse:TRANSition:DELTATime?

**Related Commands** [TRIGger:{A|B}:PULse:TRANSition:POLarity](#), [TRIGger:{A|B}:PULse:TRANSition:THReshold?](#)

**Arguments** <NR3> specifies the delta time, in seconds.

**Examples** TRIGGER:A:PULSE:TRANSITION:DELTATIME 15E-6 sets the delta time of the transition trigger to 15  $\mu$ s.

TRIGGER:A:PULSE:TRANSITION:DELTATIME? might return  
 :TRIGGER:A:PULSE:TRANSITION:DELTATIME 2.0000E-09, indicating that the delta time of the transition trigger is set to 2 ns.

## TRIGger:{A|B}:PULse:TRANSition:POLarity

This command sets or queries the polarity for the transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and choosing from the Polarity pull-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition:POLarity  
 {POSITIVE|NEGATIVE|EITHER}  
 TRIGger:{A|B}:PULse:TRANSition:POLarity?

**Related Commands** [TRIGger:{A|B}:PULse:TRANSition:DELTATime](#)

<b>Arguments</b>	<p><b>POSITIVE</b> indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.</p> <p><b>NEGATIVE</b> indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.</p> <p><b>EITHER</b> indicates either positive or negative polarity.</p>
<b>Examples</b>	<p><b>TRIGGER:A:PULSE:TRANSITION:POLARITY NEGATIVE</b> sets the transition polarity to negative.</p> <p><b>TRIGGER:A:PULSE:TRANSITION:POLARITY?</b> might return <b>:TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER</b>, indicating that the polarity can be either positive or negative.</p>

## **TRIGger:{A|B}:PULse:TRANSition:POLArity:CH<x>**

This command sets or queries the polarity for the A or B pulse transition trigger for the channel.

This command is equivalent to selecting Transition Setup from the Trig menu and then choosing from the Polarity pull-down list for the channel.

<b>Group</b>	Trigger
<b>Syntax</b>	<p><b>TRIGger:{A B}:PULse:TRANSition:POLArity:CH&lt;x&gt;</b>  <b>{EITHER NEGATIVE POSITIVE}</b>  <b>TRIGger:{A B}:PULse:TRANSition:POLArity:CH&lt;x&gt;?</b></p>
<b>Related Commands</b>	<a href="#"><b>TRIGger:{A B}:PULse:TRANSition:DELTAime</b></a>
<b>Arguments</b>	<p><b>EITHER</b> indicates either positive or negative polarity.</p> <p><b>NEGATIVE</b> indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.</p> <p><b>POSITIVE</b> indicates that a pulse edge must traverse from the lower (most negative) to higher (post positive) level for transition triggering to occur.</p>
<b>Examples</b>	<p><b>TRIGGER:A:PULSE:TRANSITION:DELTAime NEGATIVE</b> sets the trigger A transition polarity to negative.</p> <p><b>TRIGGER:A:PULSE:TRANSITION:POLARITY?</b> might return <b>:TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER</b>, indicating that the transition can be either positive or negative for trigger A.</p>

## TRIGger:{A|B}:PULse:TRANSition:QUALify

This command sets or queries the Transition Time Trigger qualification.

This is equivalent to selecting Transition Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Violation drop-down list box.

---

**NOTE.** *If you use the LOGIC argument, you must specify the channel logic conditions.*

*BUS is same as OccursAnd in the user interface.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:TRANSition:QUALify {OCCurs LOGIC BUS} TRIGger:{A B}:PULse:TRANSition:QUALify?
<b>Related Commands</b>	TRIGger:QUALification:BUS:FORMat, TRIGger:QUALification:BUS:SOUrce, TRIGger:QUALification:BUS:VALue
<b>Arguments</b>	<p>OCCurs specifies a trigger if any violations occur.</p> <p>LOGIC specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the TRIGger:{A B}:LOGIC:INPut:CH&lt;x&gt; and TRIGger:{A B}:LOGIC:THReshold:CH&lt;x&gt; commands.</p> <p>BUS specifies a trigger if the bus input qualifications meet the pattern set by the TRIGger:QUALification:BUS:VALue command.</p>
<b>Examples</b>	<p>TRIGGER:A:PULSE:TRANSITION:QUALIFY OCCURS sets the trigger qualifier to OCCurs, which initiates an A trigger if the instrument detects any transition violations.</p> <p>TRIGGER:A:PULSE:TRANSITION:QUALIFY? might return :TRIGGER:A:PULSE:TRANSITION:QUALIFY OCCURS, indicating that an A trigger will occur if the instrument detects any transition violations.</p>

## TRIGger:{A|B}:PULse:TRANSition:THReshold? (Query Only)

This query-only command returns the upper and lower threshold limits for the transition time trigger. This command is equivalent to selecting Transition Setup from the Trig menu and viewing the Upper Level and Lower Level voltage settings.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:TRANSition:THReshold?
<b>Related Commands</b>	TRIGger:{A B}:PULse:TRANSition:DELTATime, TRIGger:{A B}:PULse:TRANSition:POLarity
<b>Examples</b>	TRIGGER:A:PULSE:TRANSITION:THRESHOLD? might return :TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 1.2000;LOW 800.0000E-3, indicating that the upper threshold limit to 1.2 V and lower threshold limit for the transition time trigger to 0.8 V.

## TRIGger:{A|B}:PULse:TRANSition:THReshold:BOTH (No Query Form)

This command (no query form) sets the upper and lower thresholds for the transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and setting the desired Upper Level and Lower Level voltages.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:TRANSition:THReshold:BOTH {TTL ECL}
<b>Related Commands</b>	TRIGger:{A B}:PULse:TRANSition:THReshold:HIGH, TRIGger:{A B}:PULse:TRANSition:THReshold:LOW
<b>Arguments</b>	TTL sets the upper and lower threshold to the nominal TTL voltage levels. ECL sets the upper and the lower threshold to the nominal ECL voltage levels.
<b>Examples</b>	TRIGGER:A:PULSE:TRANSITION:THRESHOLD:BOTH TTL sets the thresholds of the transition trigger to the nominal TTL voltage levels.

## TRIGger:{A|B}:PULse:TRANSition:THReshold:HIGH

This command sets or queries the upper (most positive) transition trigger threshold. This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Upper Level voltage.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition:THReshoLd:HIGH <NR3>  
 TRIGger:{A|B}:PULse:TRANSition:THReshoLd:HIGH?

**Related Commands** [TRIGger:{A|B}:PULse:TRANSition:THReshoLd:LOW](#)

**Arguments** <NR3> specifies the upper threshold, in volts.

**Examples** TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 120E-3 sets the upper limit of the pulse transition trigger to 120 mV.

TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH? might return  
 :TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 2.0000E+00,  
 indicating that the upper limit of the pulse transition trigger is 2 V.

## TRIGger:{A|B}:PULse:TRANSition:THReshoLd:LOW

This command sets or queries the lower (most negative) transition trigger threshold. This command is equivalent to selecting Transition Setup from the Trig menu and setting the desired Lower Level voltage.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition:THReshoLd:LOW <NR3>  
 TRIGger:{A|B}:PULse:TRANSition:THReshoLd:LOW?

**Related Commands** [TRIGger:{A|B}:PULse:TRANSition:THReshoLd:HIGH](#)

**Arguments** <NR3> specifies the lower threshold, in volts.

**Examples** TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW 20E-3 sets the lower limit of the transition trigger to 20 mV.

TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW? might return  
 :TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW 50.0000E-03,  
 indicating that the lower limit of the transition trigger is 50 mV.

## TRIGger:{A|B}:PULse:TRANSition:WHEn

This command sets or queries whether to check for a transitioning signal that is faster or slower than the specified delta time. This is equivalent to selecting



Transition Setup from the Trig menu and choosing the Trigger When Transition Time setting.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:TRANSition:WHEN {FASTERthan|SLOWERthan}  
TRIGger:{A|B}:PULse:TRANSition:WHEN?

**Related Commands** [TRIGger:{A|B}:PULse:TRANSition:DELTA](#)Time, [TRIGger:{A|B}:PULse:TRANSition:POLarity](#), [TRIGger:{A|B}:PULse:TRANSition:THReshold?](#)

**Arguments** FASTERthan sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

**Examples** TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

TRIGGER:A:PULSE:TRANSITION:WHEN? might return  
:TRIGGER:A:PULSE:TRANSITION:WHEN FASTERthan, indicating that the transition triggers when the transitioning signal is faster than the set volts/second rate.

## TRIGger:{A|B}:PULse:WIDth? (Query Only)

This query-only command returns the width parameters for the pulse width trigger. This command is equivalent to selecting Width Setup from the Trig menu and then viewing the current pulse width trigger Lower Limit, Upper Limit, Trig When and Polarity settings.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WIDth?

**Related Commands** [TRIGger:{A|B}:PULse:WIDth:HIGHLimit](#), [TRIGger:{A|B}:PULse:WIDth:LOWLimit](#), [TRIGger:{A|B}:PULse:WIDth:POLarity](#), [TRIGger:{A|B}:PULse:WIDth:WHEn](#)

**Examples** TRIGGER:A:PULSE:WIDTH? might return  
:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9; HIGHLIMIT  
2.0000E-9;WHEN WITHIN; POLARITY POSITIVE as the current A width  
trigger parameters.

## TRIGger:{A|B}:PULse:WIDth:HIGHLimit

This command sets or queries the upper limit for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and setting the Upper Limit.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WIDth:HIGHLimit <NR3>  
TRIGger:{A|B}:PULse:WIDth:HIGHLimit?

**Related Commands** [TRIGger:{A|B}:PULse:WIDth?](#)

**Arguments** <NR3> specifies the width trigger upper limit in seconds.

**Examples** TRIGGER:A:PULSE:WIDTH:HIGHLIMIT 5.0E-6 sets the upper limit of the width trigger to 5  $\mu$ s.

TRIGGER:A:PULSE:WIDTH:HIGHLIMIT? might return  
:TRIGGER:A:PULSE:WIDTH:HIGHLIMIT 2.0000E-9, indicating  
that the upper limit of the width trigger is set to 2 ns.

## TRIGger:{A|B}:PULse:WIDth:LOWLimit

This command sets or queries the lower limit for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and setting the pulse Lower Limit.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WIDth:LOWLimit <NR3>  
TRIGger:{A|B}:PULse:WIDth:LOWLimit?

**Related Commands** [TRIGger:{A|B}:PULse:WIDth?](#)

**Arguments** <NR3> specifies the A pulse width trigger lower limit, in seconds.

**Examples** TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0E-6 sets the lower limit of the width trigger to 2  $\mu$ s.

TRIGGER:A:PULSE:WIDTH:LOWLIMIT? might return  
:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 1.0000E-9, indicating that the lower limit of the width trigger is set to 1 ns.

## TRIGger:{A|B}:PULse:WIDth:POLarity

This command sets or queries the polarity for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and selecting the Polarity.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WIDth:POLarity {NEGative|POSITivE}  
TRIGger:{A|B}:PULse:WIDth:POLarity?

**Related Commands** [TRIGger:{A|B}:PULse:WIDth?](#)

**Arguments** NEGative specifies a negative pulse.  
POSITivE specifies a positive pulse.

**Examples** TRIGGER:A:PULSE:WIDTH:POLARITY NEGATIVE sets the pulse polarity to negative.

TRIGGER:A:PULSE:WIDTH:POLARITY? might return  
:TRIGGER:A:PULSE:WIDTH:POLARITY POSITIVE, indicating that the pulse polarity is set to positive.

## TRIGger:{A|B}:PULse:WIDth:POLarity:CH<x>

This command sets or queries the polarity for the A or B pulse width trigger for the channel.

This command is equivalent to selecting Width Setup from the Trig menu and then selecting the pulse width trigger Polarity for the channel.

**Group** Trigger

<b>Syntax</b>	TRIGger:{A B}:PULse:WIDth:POLArity:CH<x> {NEGAtive POSITIVe} TRIGger:{A B}:PULse:WIDth:POLArity:CH<x>?
<b>Related Commands</b>	TRIGger:{A B}:PULse:WIDth?
<b>Arguments</b>	NEGAtive specifies a negative pulse. POSITIVe specifies a positive pulse.
<b>Examples</b>	TRIGGER:A:PULSE:WIDTH:POLARITY:CH4 NEGATIVE sets the pulse polarity to negative for Channel 4.  TRIGGER:A:PULSE:WIDTH:POLARITY:CH4? might return :TRIGGER:A:PULSE:WIDTH:POLARITY:CH4 POSITIVE, indicating that the pulse polarity is set to positive for Channel 4.

## TRIGger:{A|B}:PULse:WIDth:QUALify

This command sets or queries the Width Trigger qualification.

This is equivalent to selecting Width Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Width drop-down list box.

---

**NOTE.** If you use the LOGIC argument, you must specify the channel logic conditions.

BUS is same as OccursAnd in the user interface.

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:WIDth:QUALify {OCCurs LOGIC BUS} TRIGger:{A B}:PULse:WIDth:QUALify?
<b>Related Commands</b>	TRIGger:QUALification:BUS:FORMat, TRIGger:QUALification:BUS:SOUrce, TRIGger:QUALification:BUS:VALue
<b>Arguments</b>	OCCURS specifies a trigger if any detectable event occurs.  LOGIC specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the TRIGger:{A B}:LOGIC:INPut:CH<x> and TRIGger:{A B}:LOGIC:THReshold:CH<x> commands.

BUS specifies a trigger if the bus input qualifications meet the pattern set by the [TRIGger:QUALification:BUS:VALue](#) command.

**Examples** TRIGGER:A:PULSE:WIDTH:QUALIFY OCCURS sets the A trigger qualifier to OCCURS, which initiates a trigger if the instrument detects any width trigger qualifications.

TRIGGER:A:PULSE:WIDTH:QUALIFY? might return  
:TRIGGER:A:PULSE:WIDTH:QUALIFY OCCURS, indicating that an A trigger will occur if the instrument detects any of the width qualifications.

## TRIGger:{A|B}:PULse:WIDth:WHEn

This command sets or queries whether to trigger on a pulse width that falls outside (or within) the specified range of limits. You can define or query trigger pulse width upper and lower limits using the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit commands.

This command is equivalent to selecting Width Setup from the Trig menu and then choosing from the Trig When drop-down list.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WIDth:WHEn {OUTside|WITHin}  
TRIGger:{A|B}:PULse:WIDth:WHEn?

**Related Commands** [TRIGger:{A|B}:PULse:WIDth:HIGHLimit](#), [TRIGger:{A|B}:PULse:WIDth:LOWLimit](#)

**Arguments** OUTside argument causes a trigger event the duration of the pulse is greater than the high limit or less than the low limit specified. The high and low limits are specified with the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit commands respectively.

WITHin argument causes a trigger event when the duration of the pulse is within the high and low limits. The high and low limits are specified with the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit command respectively.

**Examples** TRIGGER:A:PULSE:WIDTH:WHEN WITHIN specifies that the duration of the A pulse will fall within defined high and low limits.

TRIGGER:A:PULSE:WIDTH:WHEN? might return  
:TRIGGER:A:PULSE:WIDTH:WHEN OUTSIDE, indicating the conditions for  
generating a width trigger.

## TRIGger:{A|B}:PULse:WINDow? (Query Only)

This query-only command returns the current window trigger parameters.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:WINDow?
<b>Examples</b>	<p>TRIGGER:A:PULSE:WINDOW? might return :TRIGGER:A:PULSE:WINDOW:TYPE INSIDE; WHEN OCCURS;WIDTH 2.0000e-9; LOGIC:INPUT:CH1 HIGH;CH2 X; CH3 X; CH4 LOW; :TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1 1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000; :TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW 800.0000E-3.</p>

## TRIGger:{A|B}:PULse:WINDow:EVENT

This command sets or queries the window trigger event. This command is equivalent to selecting Window Setup from the Trig menu and selecting from the Window Event box.

<b>Group</b>	Trigger
<b>Syntax</b>	<p>TRIGger:{A B}:PULse:WINDow:EVENT {OUTSIDEGreater INSIDEGreater ENTERSwindow EXITSwindow} TRIGger:{A B}:PULse:WINDow:EVENT?</p>
<b>Arguments</b>	<p>OUTSIDEGreater specifies a trigger event when the signal leaves the window defined by the threshold levels for the time specified by Width.</p> <p>INSIDEGreater specifies a trigger event when the signal enters the window defined by the threshold levels for the time specified by Width.</p> <p>ENTERSwindow specifies a trigger event when the signal enters the window defined by the threshold levels.</p>

EXITSWINDOW specifies a trigger event when the signal leaves the window defined by the threshold levels.

**Examples** TRIGGER:A:PULSE:WINDOW:WHEN OCCURS sets the window violation qualifier to OCCURS, which causes a trigger to occur when the instrument detects any window violation.

TRIGGER:A:PULSE:WINDOW:WHEN? might return  
:TRIGGER:A:PULSE:WINDOW:WHEN OCCURS, indicating that a trigger will occur if the instrument detects any window violation.

## TRIGger:{A|B}:PULse:WINDow:POLarity

This command sets or queries the pulse trigger window polarity of the selected trigger Source. The instrument triggers when the signal exits the window. This command is available only when the Window Event is set to Inside > t and not available for the rest of the window events. The logic selection is available only when the polarity is set to Either.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WINDow:POLarity  
{EITHER|NEGATIVE|POSITIVE}  
TRIGger:{A|B}:PULse:WINDow:POLarity?

**Arguments** EITHER specifies positive or negative polarity.  
NEGATIVE specifies a negative polarity.  
POSITIVE specifies a positive polarity.

**Examples** TRIGGER:A:PULSE:WINDOW:POLARITY POSITIVE sets the window polarity to positive.

TRIGGER:A:PULSE:WINDOW:POLARITY? might return  
:TRIGGER:A:PULSE:WINDOW:POLARITY EITHER, indicating that the window polarity is either positive or negative.

## TRIGger:{A|B}:PULse:WINDow:POLarity:CH<x>

This command sets or queries the pulse trigger window polarity of the specified channel. The oscilloscope triggers only when the signal exits the window. This command is available only when the Window Event is set to Inside > t and not

available for the rest of the window events. The logic selection is available only when the polarity is set to Either. <x> is the search number or channel number.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:WINDow:POLArity:CH<x> {EITHer NEGATive POSITIVe} TRIGger:{A B}:PULse:WINDow:POLArity:CH<x>?
<b>Arguments</b>	EITHer specifies positive or negative polarity. NEGATive specifies a negative polarity. POSITIVe specifies a positive polarity.
<b>Examples</b>	TRIGGER:A:PULSE:WINDOW:POLARITY:CH1 POSITIVE sets the window polarity to positive.  TRIGGER:A:PULSE:WINDOW:POLARITY:CH1? might return :TRIGGER:A:PULSE:WINDOW:POLARITY:CH1 EITHER, indicating that the window polarity is either positive or negative.

## TRIGger:{A|B}:PULse:WINDow:QUALify

This command sets or queries the Window Trigger qualification. This is equivalent to selecting Window Setup from the Trig menu and selecting Occurs, Logic, or Bus in the Trigger if Window drop-down list box.

---

**NOTE.** If you use the LOGic argument, you must specify the channel logic conditions.

*BUS is same as OccursAnd in the user interface.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:PULse:WINDow:QUALify {OCCurs LOGic BUS} TRIGger:{A B}:PULse:WINDow:QUALify?
<b>Related Commands</b>	<a href="#">TRIGger:QUALification:BUS:FORMat</a> , <a href="#">TRIGger:QUALification:BUS:SOUrce</a> , <a href="#">TRIGger:QUALification:BUS:VALue</a>



- Arguments** OCCURS specifies a trigger if any detectable event occurs.
- LOGIC specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the `TRIGger:{A|B}:LOGic:INPut:CH<x>` and `TRIGger:{A|B}:LOGic:THReshold:CH<x>` commands.
- BUS specifies a trigger if the bus input qualifications meet the pattern set by the `TRIGger:QUALification:BUS:VALue` command.
- Examples** `TRIGGER:A:PULSE:WINDOW:QUALIFY OCCURS` sets the trigger qualifier to OCCURS, which causes an A trigger to occur if the instrument detects any of the window trigger qualifications.
- `TRIGGER:A:PULSE:WINDOW:QUALIFY?` might return `:TRIGGER:A:PULSE:WINDOW:QUALIFY OCCURS`, indicating that an A trigger will occur if the instrument detects any of the window qualifications.

### TRIGger:{A|B}:PULse:WINDow:THReshold? (Query Only)

This query-only command returns the upper and lower thresholds for the window trigger. This command query is equivalent to selecting Window Setup from the Trig menu and viewing the window trigger Upper Level and Lower Level settings.

- Group** Trigger
- Syntax** `TRIGger:{A|B}:PULse:WINDow:THReshold?`
- Examples** `TRIGGER:A:PULSE:WINDOW:THRESHOLD?` might return `:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW 800.0000E-3`, indicating that the upper threshold is 1.2 V and that the lower threshold is 800 mV.

### TRIGger:{A|B}:PULse:WINDow:THReshold:BOTH (No Query Form)

This command (no query form) sets the upper and lower switching thresholds for the window trigger. This command is equivalent to selecting Window Setup from the Trig menu and then setting the Upper Level and Lower Level voltages.

- Group** Trigger
- Syntax** `TRIGger:{A|B}:PULse:WINDow:THReshold:BOTH {TTL|ECL}`

**Arguments** TTL argument sets the upper and the lower threshold to the nominal TTL voltage levels.

ECL argument sets the upper and the lower threshold to the nominal ECL voltage levels.

**Examples** TRIGGER:A:PULSE:WINDOW:THRESHOLD:BOTH TTL sets the threshold of the window trigger to the nominal TTL voltage levels.

## TRIGger:{A|B}:PULse:WINDow:THReshold:HIGH

This command sets or queries the upper limit for the pulse window trigger. This command is equivalent to selecting Window Setup from the Trig menu and setting the window trigger Upper Level voltage.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WINDow:THReshold:HIGH <NR3>  
TRIGger:{A|B}:PULse:WINDow:THReshold:HIGH?

**Arguments** <NR3> specifies the threshold value in volts.

**Examples** TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 2.0, sets the upper limit of the pulse window trigger to 2 V.

TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH? might return  
:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.1000E+00, indicating that the upper limit of the window trigger is set to 1.1 V.

## TRIGger:{A|B}:PULse:WINDow:THReshold:LOW

This command sets or queries the lower limit for the window trigger. This command is equivalent to selecting Window Setup from the Trig menu and setting the Lower Level voltage.

**Group** Trigger

**Syntax** TRIGger:{A|B}:PULse:WINDow:THReshold:LOW <NR3>  
TRIGger:{A|B}:PULse:WINDow:THReshold:LOW?

<b>Arguments</b>	<NR3> specifies the threshold value in volts.
<b>Examples</b>	<p>TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW 0.8 sets the lower threshold of the window trigger to 0.8 V.</p> <p>TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW? might return :TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW 1.2000E-01, indicating that the lower limit of the window trigger is currently set to 120 mV.</p>

## TRIGger:{A|B}:PULse:WINDow:WIDTH

This command sets or queries the minimum width for a window violation. This command is equivalent to selecting Window Setup from the Trig menu, selecting Wider than in the Trigger When box, and setting the Width.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:{A B}:PULse:WINDow:WIDTH &lt;NR3&gt; TRIGger:{A B}:PULse:WINDow:WIDTH?</pre>
<b>Arguments</b>	<NR3> argument specifies the minimum width in seconds.
<b>Examples</b>	<p>TRIGGER:A:PULSE:WINDOW:WIDTH 15E-9 sets the minimum width of a valid window violation to 15 ns.</p> <p>TRIGGER:A:PULSE:WINDOW:WIDTH? might return :TRIGGER:A:PULSE:WINDow:WIDTH 2.0000E-09 indicating that the minimum width of a valid window violation is 2 ns.</p>

## TRIGger:{A|B}:READY? (Query Only)

This command queries the trigger ready state and provides the immediate state from the trigger system. It is a more synchronous means of determining when the oscilloscope is ready to trigger. The [TRIGger:STATE?](#) reflects a less-frequently updated status of the trigger LEDs on the instrument front panel.

---

**NOTE.** *A delay or B trigger can become 'ready' under different circumstances. If delay by time is enabled, the B trigger becomes ready after the "trigger after time" has elapsed. If delay by events is enabled, the B trigger becomes ready after the A trigger has occurred. Also, a 'ready' state cannot be guaranteed when Reset Events are used in an A to B Event Sequence trigger.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:READY?
<b>Related Commands</b>	<a href="#">TRIGger:STATE?</a> , <a href="#">TRIGger:B:BY</a> , <a href="#">TRIGger:B:TIME</a> , <a href="#">TRIGger:B:EVENTS:COUNT</a>
<b>Examples</b>	<p>TRIGGER:{A B}:READY? might return :TRIGGER:{A B}:READY 1, indicating that all pretrigger information has been acquired and that the instrument is ready to accept a trigger</p> <p>TRIGGER:{A B}:READY? might return :TRIGGER:{A B}:READY 0, indicating that the instrument is not yet ready to accept a trigger</p>

## TRIGger:{A|B}:TYPE

This command sets or queries the type of A or B trigger. Logic and Pulse triggers contain classes. Logic triggers consist of State and Pattern classes; Pulse triggers consist of Glitch, Runt, Width, Transition, Timeout, and Window classes. Once you have set the trigger type, you may also need to identify the associated trigger class. For details on selecting Logic and Pulse trigger classes, see [TRIGger:{A|B}:LOGIC:CLASS](#) and [TRIGger:{A|B}:PULSE:CLASS](#) respectively. This command is similar to selecting Event Trigger Setup from the Trig menu and then selecting the desired Trigger Type.

Some trigger types are not available on some instruments.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:{A B}:TYPE {EDGE LOGIC PULSE VIDEO I2C CAN SPI COMMUNICATION SERIAL RS232}

---

**NOTE.** Arguments *EDGE*, *LOGIC*, *PULSE*, *VIDEO*, *I2C*, *CAN*, *SPI*, *COMMUNICATION*, *SERIAL*, or *RS232* are supported by A trigger events.

---



---

**NOTE.** Arguments *EDGE*, *LOGIC*, or *PULSE* are supported by B trigger events.

---

TRIGger:{A|B}:TYPE?

**Related Commands** [TRIGger:{A|B}:EDGE?](#), [TRIGger:{A|B}:LOGic:CLAss](#), [TRIGger:{A|B}:PULse:CLAss](#)

**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.

**LOGic** specifies that a trigger occurs when specified conditions are met and is controlled by the `TRIGger:A:LOGic` commands.

**PULse** specifies that a trigger occurs when a specified pulse is found and is controlled by the `TRIGger:A:PULse` commands.

**VIDeo** specifies that the trigger occurs when a video signal is found. Requires an instrument with video hardware.

**I2C** specifies that a trigger occurs when an Inter-IC Control signal is found.

**CAN** specifies that a trigger occurs when a Controller Area Network frame signal is found.

**SPI** specifies that a trigger occurs when a Serial Peripheral Interface signal is found.

**COMMunication** (Option MTM) specifies that a trigger occurs when a communications signal is found. Supports AMI, HDB3, B3ZS, B6ZS, B8ZS, CMI, MLT3, Manchester, and NRZ encoded communications signals. **COMMunication** is available only if Option MTM is installed.

**SERIAL** specifies that a trigger occurs when NRZ-encoded data is found, providing a 32-bit serial word. This argument is available with instruments with Option PTM.

**RS232** takes a signal on a data source and allows you to trigger on data within the RS232 bitstream. The data is only one byte wide.

**Examples**

`TRIGGER:A:TYPEEDGE` sets the A trigger type to EDGE.

`TRIGGER:A:TYPE?` might return `:TRIGGER:A:TYPE PULSE` indicating that the A trigger type is a pulse trigger.

## **TRIGger:{A|B}:UPPerthreshold:CH<x>**

This command sets or queries the CH<x> upper trigger level for `TRIGger:LVLsrcpreference SRCDependent`. The CH<x> range is 1 to 4.

**Group** Trigger

**Syntax** TRIGger:{A|B}:UPPERthreshold:CH<x> {ECL | TTL | <NR3>}  
 TRIGger:{A|B}:UPPERthreshold:CH<x>?

**Related Commands** [TRIGger:LVLsrcpreference](#), [TRIGger:{A|B}:LOWerthreshold:CH<x>](#)

**Arguments** ECL specifies the ECL high level.  
 TTL specifies the TTL high level.  
 <NR3> specifies the trigger level in user units (usually volts).

**Examples** TRIGGER:A:UPPERTHRESHOLD:CH1 TTL This command sets the A trigger level for Channel 1 to TTL high 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

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 A 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?

**Related Commands** [TRIGger:{A|B}:EDGE?](#), [TRIGger:{A|B}:LOGic?](#), [TRIGger:{A|B}:PULse?](#)

**Arguments** SETLevel sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

**Examples** TRIGGER:ASETLEVEL sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.  
 TRIGGER:A? might return the following trigger parameters:  
 :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;;TRIGGER:A:LOGIC:CLASS
PATTERN;FUNCTION AND;THRESHOLD:CH1 1.2000;CH2
1.2000;CH3 1.2000;CH4 1.2000;;TRIGGER:A:LOGIC:INPUT:CH1
HIGH;CH2 X;CH3 X;;TRIGGER:A:LOGIC:PATTERN:INPUT:CH4
X;;TRIGGER:A:LOGIC:PATTERN:WHEN
TRUE;WHEN:LESSLIMIT 5.0000E-9;MORELIMIT
5.0000E-9;;TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD
1.2000;SOURCE CH2;;TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD
1.2000;SOURCE CH1;;TRIGGER:A:LOGIC:SETHOLD:HOLDTIME
2.0000E-9;SETTIME 3.0000E-9;;TRIGGER:A:LOGIC:STATE:INPUT:CH4
RISE;;TRIGGER:A:LOGIC:STATE:WHEN
TRUE;;TRIGGER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;POLARITY
POSITIVE;;TRIGGER:A:PULSE:WINDOW:TYPE INSIDE;WHEN
OCCURS;WIDTH 2.0000E-9;LOGIC:INPUT:CH1 HIGH;CH2 HIGH;CH3
HIGH;CH4 HIGH;;TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;;TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;;TRIGGER:A:PULSE:RUNT:POLARITY
POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH
2.0000E-9;;TRIGGER:A:PULSE:TRANSITION:DELTA TIME
2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;TRIGGER:A:PULSE:TRANSITION:WHEN
FASTER THAN;;TRIGGER:A:PULSE:WIDTH:LOWLIMIT
2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;POLARITY
POSITIVE;;TRIGGER:A:PULSE:TIMEOUT:POLARITY
STAYSHIGH;TIME 2.0000E-9;;TRIGGER:A:VIDEO:CUSTOM:FORMAT
INTERLACED;SCAN RATE1;;TRIGGER:A:VIDEO:FIELD
ALLFIELDS;HOLD OFF:FIELD 0.0000;;TRIGGER:A:VIDEO:LINE
1;POLARITY NORMAL;SCAN RATE1;SOURCE CH1;STANDARD
NTSC;;TRIGGER:A:COMMUNICATION:STANDARD CUSTOM;CODE
NRZ;BITRATE 1544000;AMI:PULSEFORM PLUSONE;THRESHOLD:HIGH
0.0000;LOW 0.0000;;TRIGGER:A:COMMUNICATION:CLOCK:POLARITY
RISE;;TRIGGER:A:COMMUNICATION:CMI:PULSEFORM
PLUSONE;;TRIGGER:A:COMMUNICATION:SOURCE CH1;SOURCE:TYPE DATA.

```

## TRIGger:A:BUS:DATA:FORMat

This command sets or queries the trigger data format for the bus.

**Group** Trigger

**Syntax** TRIGger:A:BUS:DATA:FORMat {BINary|HEXadecimal}  
TRIGger:A:BUS:DATA:FORMat?

**Arguments** BINary sets the trigger data format to binary.  
HEXadecimal sets the trigger data format to hexadecimal.

**Examples** TRIGGER:A:BUS:DATA:FORMAT HEX sets the trigger data format for the bus to hexadecimal.

TRIGGER:A:BUS:DATA:FORMAT? might return  
:TRIGGER:A:BUS:DATA:FORMAT BIN, indicating that the trigger data format for the bus is set to binary.

## TRIGger:A:BUS:DATA:VALue

This command sets or queries the trigger data value depending on the format selected for the bus. The data value varies depending on the bus type and the number of input signals.

**Group** Trigger

**Syntax** TRIGger:A:BUS:DATA:VALue <QString>  
TRIGger:A:BUS:DATA:VALue?

**Arguments** <QString> is the data value in binary or hexadecimal format. The valid characters are 0-9, A-F, and X for addresses in hexadecimal format and 0, 1, and X otherwise.

**Examples** TRIGGER:A:BUS:DATA:VALUE "656565XX65656AF37565" sets the trigger data value for the bus to 656565XX65656AF37565 in hexadecimal format.

TRIGGER:A:BUS:DATA:VALUE? might return :TRIGGER:A:BUS:DATA:VALUE "0100X0XX101", indicating that the trigger data value for the value is set to 0100X0XX101 in binary format.

## TRIGger:A:BUS:I2C:ADDRess:FORMat

This command sets or queries the address format for the I2C bus.

**Group** Trigger



<b>Syntax</b>	TRIGger:A:BUS:I2C:ADDRESS:FORMAT {BINary HEXadecimal} TRIGger:A:BUS:I2C:ADDRESS:FORMAT?
<b>Arguments</b>	BINary specifies the address format for the I2C bus to binary. HEXadecimal specifies the address format for the I2C bus to hexadecimal.
<b>Examples</b>	TRIGGER:A:BUS:I2C:ADDRESS:FORMAT BIN sets the address format for the I2C bus to binary.  TRIGGER:A:BUS:I2C:ADDRESS:FORMAT? might return :TRIGGER:A:BUS:I2C:ADDRESS:FORMAT HEX, indicating that the address format for the I2C bus is set to hexadecimal.

## TRIGger:A:BUS:I2C:ADDRESS:MODE

This command sets or queries the address mode for the I2C bus.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:BUS:I2C:ADDRESS:MODE {7Bit 10Bit} TRIGger:A:BUS:I2C:ADDRESS:MODE?
<b>Arguments</b>	7Bit specifies the 7bit I2C address mode. 10Bit specifies the 10bit I2C address mode.
<b>Examples</b>	TRIGGER:A:BUS:I2C:ADDRESS:MODE 7Bit sets the I2C address mode for the bus to 7 bit.  TRIGGER:A:BUS:I2C:ADDRESS:MODE? might return :TRIGGER:A:BUS:I2C:ADDRESS:MODE 7BIT, indicating that the I2C address mode for the bus to 7 bit.

## TRIGger:A:BUS:I2C:ADDRESS:VALue

This command sets or queries the binary address string used for the I2C trigger of the bus.

<b>Group</b>	Trigger
--------------	---------

**Syntax** TRIGger:A:BUS:I2C:ADDRESS:VALUE <QString>  
 TRIGger:A:BUS:I2C:ADDRESS:VALUE?

**Arguments** <QString> specifies the address value. This is either 7 bits or 10 bits 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** TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "011XX11" sets the 7-bit address value to 011XX11.

TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE? might return  
 :TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "A45X1CX", indicating that the 7 bit address value is set to A45X1CX.

## TRIGger:A:BUS:I2C:CONDition

This command sets or queries the trigger condition for the I2C bus.

---

**NOTE.** This command is available only on MSO70000 Series products.

---

**Group** Trigger

**Syntax** TRIGger:A:BUS:I2C:CONDition  
 {START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}  
 TRIGger:A:BUS:I2C:CONDition?

**Arguments** START sets the trigger condition for the I2C bus to start.

STOP sets the trigger condition for the I2C bus to stop.

REPEATstart sets the trigger condition for the I2C bus to repeat of start.

ACKMISS sets the trigger condition for the I2C bus to missing of acknowledgement.

ADDRESS sets the trigger condition for the I2C bus to address.

DATA sets the trigger condition for the I2C bus to data.

ADDRANDDATA sets the trigger condition for the I2C bus to address and data.

**Examples** TRIGGER:A:BUS:B1:I2C:CONDITION START sets the trigger condition for the I2C bus to start.

TRIGGER:A:BUS:B1:I2C:CONDITION? might return  
:TRIGGER:A:BUS:B1:I2C:CONDITION STOP, indicating that the trigger  
condition for the I2C bus is set to stop.

## TRIGger:A:BUS:I2C:DATA:DIRection

This command sets or queries the data direction for the I2C bus.

---

**NOTE.** *This command is available only on MSO70000 Series products.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:BUS:I2C:DATA:DIRection {{READ WRITE EITHER}} TRIGger:A:BUS:I2C:DATA:DIRection?
<b>Arguments</b>	<p>READ sets the data direction for the I2C bus to read.</p> <p>WRITE sets the data direction for the I2C bus to write.</p> <p>EITHER sets the data direction for the I2C bus to either read or write.</p>
<b>Examples</b>	<p>TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION WRITE sets data direction for the I2C bus to write.</p> <p>TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION? might return TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION READ, indicating that data direction for the I2C bus is set to read.</p>

## TRIGger:A:BUS:SOUrce

This command sets or returns the source for a bus trigger.

---

**NOTE.** *This command is available only on MSO70000 Series products.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	<p>TRIGger:A:BUS:SOUrce {A11 BUS1 BUS2 BUS3 BUS4 BUS5 BUS6 BUS7 BUS8  BUS9 BUS10 BUS11 BUS12 BUS13 BUS14 BUS15 BUS16} TRIGger:A:BUS:SOUrce?</p>

- Arguments** All sets the source to all.  
 BUS1–BUS16 sets the selected source to the bus.
- Examples** TRIGGER:A:BUS:SOURCE BUS2 sets the selected source for the BUS2 trigger.  
 TRIGGER:A:BUS:SOURCE? might return TRIGGER:A:BUS:SOURCE BUS1,  
 indicating that the selected source for the bus trigger is set to BUS1.

## TRIGger:A:BUS:SPI:CONDition

This command sets or queries the trigger condition for the SPI bus.

---

**NOTE.** *This command is available only on MSO70000 Series products.*

---

- Group** Trigger
- Syntax** TRIGger:A:BUS:SPI:CONDition {SS|DATA}  
 TRIGger:A:BUS:SPI:CONDition?
- Arguments** SS sets the trigger condition to Slave selection.  
 DATA sets the trigger condition to Master-In Slave-Out and Master-Out Slave-In.
- Examples** TRIGGER:A:BUS:B1:SPI:CONDITION SS sets the trigger condition to Slave selection.  
 TRIGGER:A:BUS:B1:SPI:CONDITION? might return  
 :TRIGGER:A:BUS:B1:SPI:CONDITION SS, indicating that the trigger condition is set to Slave selection.

## TRIGger:A:CAN:CONDition

CAN option only: This command sets or returns the CAN condition.

- Group** Low Speed Serial Trigger
- Syntax** TRIGger:A:CAN:CONDition  
 {SOF|FRAMetype|IDENTifier|DATA|IDANDDATA|EOF|ACKMISS}  
 TRIGger:A:CAN:CONDition?

**Arguments** SOF enables triggering on the start of frame.  
 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.  
 ACKMISS enables triggering on a missing acknowledge.  
 ERROR enables triggering on an error within a frame.

**Examples** TRIGGER:A:CAN:CONDITION DATA enables triggering on matching CAN data.  
 TRIGGER:A:CAN:CONDITION? might return :TRIGGER:A:CAN:CONDITION  
 EOF, indicating that the triggering is set on the end of frame.

## TRIGGER:A:CAN:DATA:DIRection

CAN option only: This command sets or queries the CAN trigger condition to be valid on a READ, WRITE, or either.

**Group** Low Speed Serial Trigger

**Syntax** TRIGGER:A:CAN:DATA:DIRection {READ|WRITE|NOCARE}  
 TRIGGER:A: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:CAN:DATA:DIRECTION WRITE sets the CAN data direction to Write.  
 TRIGGER:A:CAN:DATA:DIRECTION? might return  
 :TRIGGER:A:CAN:DATA:DIRECTION READ, indicating that the CAN data  
 direction is set to Read.

## TRIGger:A:CAN:DATA:LEVel

CAN option only: This command sets or queries the CAN Trigger threshold for the CAN data source.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:CAN:DATA:LEVel <NR3>  
TRIGger:A:CAN:DATA:LEVel?

**Arguments** <NR3> specifies the CAN trigger data level. The level can be  $\pm 12.0$  divisions.

**Examples** TRIGGER:A:CAN:DATA:LEVEL 0.0 sets the CAN trigger data level to 0.0 Volts.  
TRIGGER:A:CAN:DATA:LEVEL? might return :TRIGGER:A:CAN:DATA:LEVEL 5.0000, indicating that the CAN trigger data level is set to 5.0 volts.

## TRIGger:A:CAN:DATA:SOUrce

CAN option only: This command sets or queries the CAN data source.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:CAN:DATA:SOUrce CH<x>  
TRIGger:A:CAN:DATA:SOUrce?

**Arguments** CH<x> specifies the source for the CAN trigger. x can be 1, 2, 3, or 4.

**Examples** TRIGGER:A:CAN:DATA:SOURCE CH1 sets the CAN trigger data source to channel CH1.  
TRIGGER:A:CAN:DATA:SOURCE? might return :TRIGGER:A:CAN:DATA:SOURCE CH4, indicating that the CAN trigger data source is set to channel CH4.

## TRIGger:A:CAN:DATA:VALue

CAN option only: This command sets or queries the binary data string used for CAN Trigger if the trigger condition is ID or IDANDDATA.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:CAN:DATA:VALUE <QString> TRIGger:A:CAN:DATA:VALUE?
<b>Arguments</b>	<QString> is up to 32 bits specifying the CAN data value.
<b>Examples</b>	TRIGGER:A:CAN:DATA:VALUE "1011" sets the CAN data value to 1011. TRIGGER:A:CAN:DATA:VALUE? might return :TRIGGER:A:CAN:DATA:SOURCE 10111, indicating that the CAN data value is set to 10111.

## TRIGger:A:CAN:FORMat

CAN option only: This command sets or queries the CAN data format.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:CAN:FORMat {BINary HEX} TRIGger:A:CAN:FORMat?
<b>Arguments</b>	BINary specifies binary as the CAN data format. HEX specifies hexadecimal as the CAN data format.
<b>Examples</b>	TRIGGER:A:CAN:FORMAT BINARY sets the CAN trigger data format to binary. TRIGGER:A:CAN:FORMAT? might return :TRIGGER:A:CAN:DATA:FORMAT HEX, indicating that the CAN data format is set to hexadecimal.

## TRIGger:A:CAN:FRAMeType

CAN option only: This command sets or queries the CAN trigger frame type.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:CAN:FRAMeType {DATA REMOte ERROR OVERLOAD} TRIGger:A:CAN:FRAMeType?

<b>Arguments</b>	DATA specifies a data frame type. REMOte specifies a remote frame type. ERROR specifies an error frame type. OVERLOAD specifies an overload frame type.
<b>Examples</b>	TRIGGER:A:CAN:FRAMETYPE DATA sets the CAN trigger frame type to data. TRIGGER:A:CAN:FRAMETYPE? might return :TRIGGER:A:CAN:DATA:FRAMETYPE REMOTE, indicating that the CAN trigger frame type is set to remote.

## TRIGger:A:CAN:IDENTifier:MODE

CAN option only: This command sets or queries the CAN trigger identifier mode.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:CAN:IDENTifier:MODE {STANDARD EXTENDED} TRIGger:A:CAN:IDENTifier:MODE?
<b>Arguments</b>	STANDARD sets the CAN trigger identifier mode to standard mode. EXTENDED sets the CAN trigger identifier mode to extended mode.
<b>Examples</b>	TRIGGER:A:CAN:IDENTIFIER:MODESTANDARD sets the CAN trigger identifier mode to standard. TRIGGER:A:CAN:IDENTIFIER:MODE? might return :TRIGGER:A:CAN:IDENTIFIER:MODE EXECUTE, indicating that the CAN trigger identifier mode is set to extended.

## TRIGger:A:CAN:IDENTifier:VALue

CAN option only: This command sets or queries the binary address string used for the CAN trigger if the trigger condition is ID or IDANDDATA.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:CAN:IDENTifier:VALue <QString> TRIGger:A:CAN:IDENTifier:VALue?



- Arguments** <QString> is up to 29 bits specifying the CAN identifier value.
- Examples** TRIGGER:A:CAN:IDENTIFIER:VALUE "1011" sets the CAN trigger identifier value to 1011.
- TRIGGER:A:CAN:IDENTIFIER:VALUE? might return  
:TRIGGER:A:CAN:IDENTIFIER:VALUE "10111", indicating that the CAN trigger identifier value is set to 10111.

## TRIGger:A:CAN:PROBE

CAN option only: This command sets or queries the probing method used to probe the CAN signal.

- Group** Low Speed Serial Trigger
- Syntax** TRIGger:A:CAN:PROBE {CANL|CANH|DIFFerential|TX|RX}  
TRIGger:A:CAN:PROBE?
- Arguments** CANL specifies that the probing method is CANL.  
CANH specifies that the probing method is CANH.  
DIFFerential specifies that the probing method used is differential.  
TX specifies that the probing method is TX.  
RX specifies that the probing method is RX.
- Examples** TRIGGER:A:CAN:PROBE CANH sets the method used to probe the CAN signal to CANH.
- TRIGGER:A:CAN:PROBE? might return TRIGGER:A:CAN:PROBE DIFFERENTIAL, indicating that the probing method is differential.

## TRIGger:A:CAN:SPEEd

CAN option only: This command sets or queries the bit rate of the CAN system.

- Group** Low Speed Serial Trigger
- Syntax** TRIGger:A:CAN:SPEEd <NR3>  
TRIGger:A:CAN:SPEEd?

**Arguments** <NR3> specifies the bit rate of the CAN system. Possible values are 1M, 800K, 500K, 250K, 125K, 100K, 83.3K, 62,5K, 50K, 33K, 20K, and 10K.

**Examples** TRIGGER:A:CAN:SPEED 33K sets the CAN systems with a bit rate of 33K.  
 TRIGGER:A:CAN:SPEED? might return TRIGGER:A:CAN:SPEED 50K, indicating that the bit rate of the CAN system is set to 50K.

## TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:PULSEForm

This entry covers five separate commands, one each for AMI and AMI subtypes. The commands set or query the AMI pulse form to one of three possibilities.

**Conditions** This command is only valid if the instrument has Option MTM is installed

**Group** Trigger

**Syntax** TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:PULSEForm  
 {PLUSOne|MINUSOne| EYEdiagram}  
 TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:PULSEForm?

**Arguments** PLUSOne corresponds to the Isolated +1 on the front panel menu.  
 MINUSOne corresponds to the Isolated -1 on the front panel menu.  
 EYEdiagram positions the trigger point off screen in order to display the eye.

**Examples** TRIGGER:A:COMMUNICATION:AMI:PULSEFORM PLUSONE sets the pulse form to PLUSONE.  
 TRIGGER:A:COMMUNICATION:AMI:PULSEFORM? might return  
 :TRIGGER:A:COMMUNICATION:AMI:PULSEFORM PLUSONE, indicating that the AMI PULSFORM is set to PLUSONE.

## TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:HIGH

This entry covers five separate commands, one each for AMI and AMI subtypes. This command sets or queries the threshold high level.

**Conditions** This command is only valid if the instrument has Option MTM is installed

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:COMMunication:{AMI HDB3 B3ZS B6ZS B8ZS}:THReshold: HIGH &lt;NR3&gt; TRIGger:A:COMMunication:{AMI HDB3 B3ZS B6ZS B8ZS}:THReshold: HIGH?</pre>
<b>Arguments</b>	<NR3> is the high threshold parameter expressed in volts.
<b>Examples</b>	<p>TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH 2.8E-2 sets the threshold high level to 2.8E-2.</p> <p>TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH? might return :TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH 2.8E-2, indicating the threshold high level in volts.</p>

## TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:LOW

This entry covers five separate commands, one each for AMI and AMI subtypes. This command sets or queries the threshold low level.

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed
<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:COMMunication:{AMI HDB3 B3ZS B6ZS B8ZS}:THReshold: LOW &lt;NR3&gt; TRIGger:A:COMMunication:{AMI HDB3 B3ZS B6ZS B8ZS}:THReshold: LOW?</pre>
<b>Arguments</b>	<NR3> specifies the high threshold parameter, expressed in volts.
<b>Examples</b>	<p>TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW 2.8E-2 sets the threshold low level to 2.8E-2.</p> <p>TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW? might return :TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW 2.8E-2, indicating the threshold low level in volts.</p>

## TRIGger:A:COMMunication:BITRate

This command sets or queries the bit rate.

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:BITRate <NR3> TRIGger:A:COMMunication:BITRate?
<b>Arguments</b>	<NR3> is a non-negative number greater than one and expressed as bits per second.
<b>Examples</b>	TRIGGER:A:COMMUNICATION:BITRATE 1.544E6 sets the bit rate to 1.544 Mb/s.  TRIGGER:A:COMMUNICATION:BITRATE? might return :TRIGGER:A:COMMUNICATION:BITRATE 1.544e6, indicating a bit rate of 1.544 Mb/s.

## TRIGger:A:COMMunication:CLOCK:POLarity

This command sets or queries the communication clock polarity

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:CLOCK:POLarity {RISe FALL} TRIGger:A:COMMunication:CLOCK:POLarity?
<b>Arguments</b>	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.
<b>Examples</b>	TRIGGER:A:COMMUNICATION:CLOCK:POLARITYRISE sets RISE as the clock polarity.  TRIGGER:A:COMMUNICATION:CLOCK:POLARITY? might return :TRIGGER:A:COMMUNICATION:CLOCK:POLARITY RISE, indicating that the polarity is to trigger on the rising edge of the clock.

## TRIGger:A:COMMunication:CMI:PULSEForm

This command sets or queries the CMI pulse form.

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:CMI:PULSEForm {PLUSone MINUSone ZERO EYEdiagram} TRIGger:A:COMMunication:CMI:PULSEForm?
<b>Arguments</b>	PLUSone triggers on a positive mark. MINUSone triggers on a negative mark. ZERO triggers on the bit representing zero. EYEdiagram positions the trigger point off screen in order to display the eye.
<b>Examples</b>	TRIGGER:A:COMMUNICATION:CMI:PULSEFORMPLUSONE sets triggering to occur on a positive mark.  TRIGGER:A:COMMUNICATION:CMI:PULSEFORM? might return :TRIGGER:A:COMMUNICATION:CMI:PULSEFORM PLUSONE, indicating that the pulse form is PLUSONE.

## TRIGger:A:COMMunication:CODE

This command sets or queries the signal code that the communications trigger should expect on the incoming signal.

<b>Conditions</b>	This command is valid only if the instrument has Option MTM is installed.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:CODE {AMI HDB3 B3ZS B6ZS B8ZS CMI NRZ MLT3 MANChester} TRIGger:A:COMMunication:CODE?
<b>Arguments</b>	AMI HDB3

B3ZS  
B6ZS  
B8ZS  
CMI  
NRZ  
MLT3  
MANChester

**Examples** TRIGGER:A:COMMUNICATION:CODE AMI sets the AMI as the line code that the communications trigger expects on the incoming signal.

TRIGGER:A:COMMUNICATION:CODE? might return  
:TRIGGER:A:COMMUNICATION:CODE AMI, indicating that the code is AMI.

## TRIGger:A:COMMunication:SOURce

This command sets or queries the source channel.

**Conditions** This command is valid only if the instrument has Option MTM is installed.

**Group** Trigger

**Syntax** TRIGger:A:COMMunication:SOURce {CH1|CH2|CH3|CH4}  
TRIGger:A:COMMunication:SOURce?

**Arguments** CH1 argument selects CH 1 as the source channel.  
CH2 argument selects CH 2 as the source channel.  
CH3 argument selects CH 3 as the source channel.  
CH4 argument selects CH 4 as the source channel.

**Examples** TRIGGER:A:COMMUNICATION:SOURCECH1 sets Channel 1 as the source channel.

TRIGGER:A:COMMUNICATION:SOURCE? might return  
:TRIGGER:A:COMMUNICATION:SOURCE CH1, indicating that the source is Channel 1.

## TRIGger:A:COMMunication:SOURce:TYPe

This command sets or queries the source type. This command works only when the Eye Diagram pulseform is selected.

<b>Conditions</b>	This command is only valid if the instrument has Option MTM is installed.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:SOURce:TYPe {DATA CLOCK RECOVERed} TRIGger:A:COMMunication:SOURce:TYPe?
<b>Arguments</b>	<p>DATA causes the instrument to trigger and shift five unit intervals to form the expected eye pattern.</p> <p>CLOCK causes the instrument to trigger but no shift occurs. The clock type causes random triggers with respect to the data channel, which must be one of the other three channels.</p> <p>RECOVERed causes the instrument to trigger on the recovered clock from the data signal attached to communication source forming an eye pattern on the source.</p>
<b>Examples</b>	<p>TRIGGER:A:COMMUNICATION:SOURCE:TYPECLOCK sets the source type to clock.</p> <p>TRIGGER:A:COMMUNICATION:SOURCE:TYPE? might return :TRIGGER:A:COMMUNICATION:SOURCE:TYPE DATA, indicating that the source type is data.</p>

## TRIGger:A:COMMunication:STANdard

This command sets or queries the standard that identifies the code and bit rate. The bit rate is used to compute the Unit Interval, which is the inverse of the bit rate. The Unit Interval influences time skew in an Eye Diagram, where you perform post processing on AMI isolated pulses, and pulse width settings if CMI.

<b>Conditions</b>	This command is valid only if the instrument has Option MTM is installed.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:COMMunication:STANdard {ATAG1 ATAG2 ATAG3 CLOCKCoax

CLOCK Symmetrical | Custom | D1 | D2 | DS0 Contra |  
 DS0 Double | DS0 Single | DS0 Timing | DS1 | DS1A |  
 DS1C | DS2 | DS2RATE Coax | DS2RATE Symmetrical |  
 DS3 | DS4NA | E1 | E2 | E3 | E4 | ENET100 | ENET1250 |  
 ENETXAUI | FC133 | FC266 | FC531 | FC1063 | FC2125 | FC4250 |  
 FST | FW1394BS400B | FW1394BS1600B | HST | INF2\_5G |  
 OC1 | OC3 | OC12 | OC48 | OC48\_FEC | PCIEXPRESS | RATE32Mbit |  
 RATE97Mbit | RIO\_500M | RIO\_750M |  
 RIO\_1G | RIO\_2G | RIO\_1\_5G | RIO\_SERIAL\_1G |  
 RIO\_SERIAL\_2G | RIO\_SERIAL\_3G | SAS1\_5 | SAS3\_? |  
 SFI5\_2 | SFI5\_3 | STM0\_CMI | STM0\_HDBX | STM1E | STS1 | STS3 |  
 TFI5\_2 | TFI5\_3 | VIDEO270 | VIDEO292M | VIDEO360 | VSROC192 }  
 TRIGGER:A:COMMUNICATION:STANDARD?

**Related Commands**    [MASK:STANDARD](#)

**Arguments**    The following table shows various standards and their corresponding parameters.

**Table 2-46: Communication Trigger Standards**

Argument	Code	Bit Rate	Mask Type	Models
ATAG1	NRZ	1.5 Gb/s	Serial ATA	≥2.5 GHz
ATAG2	NRZ	3.0 Gb/s	Serial ATA	
ATAG3	NRZ	6.0 Gb/s	Serial ATA	
CLOCK Coax	None	2.048 Mb/s	ITU-T G.703	All
CLOCK Symmetrical	None	2.048 Mb/s	ITU-T G.703	All
Custom	AMI, B3ZS, B6ZS, B8ZS, CMI, HDB3, MLT3, NRZ	1.544 Mb/s		All
D1	NRZ	270 Mb/s	Video	All
D2	NRZ	143.18 Mb/s	Video	All
DS0 Contra	AMI	16.0 kb/s	ITU-T G.730	All
DS0 Double	AMI	32.0 kb/s	ITU-T G.730	All
DS0 Single	AMI	64.0 kb/s	ITU-T G.730	All
DS0 Timing	AMI	32.0 kb/s	ITU-T G.730	All
DS1	AMI B8ZS	1.544 Mb/s	ANSI T1.102	All
DS1A	AMI HDB3	2.048 Mb/s	ANSI T1.102	All
DS1C	AMI B8ZS	3.152 Mb/s	ANSI T1.102	All
DS2	AMI B6ZS	6.312 Mb/s	ANSI T1.102	All
DS2RATE Coax	AMI B8ZS	6.312 Mb/s	ITU-T G.703	All
DS2RATE Symmetrical	AMI B6ZS	6.312 Mb/s	ITU-T G.703	All



**Table 2-46: Communication Trigger Standards (cont.)**

<b>Argument</b>	<b>Code</b>	<b>Bit Rate</b>	<b>Mask Type</b>	<b>Models</b>
DS3	AMI B3ZS	44.736 Mb/s	ANSI T1.102	All
DS4NA	CMI	139.3 Mb/s	ANSI T1.102	All
E1	AMI HDB3	2.048 Mb/s	ITU-T G.703	All
E2	AMI HDB3	8.448 MB/s	ITU-T G.703	All
E3	AMI HDB3	34.368 Mb/s	ITU-T G.703	All
E4	CMI	139.26 Mb/s	ITU-T G.703	All
ENET100	MLT3	100 Mb/s	Ethernet	All
ENET1250	NRZ	1.25 Gb/s	Ethernet	≥2.5 GHz
ENETXAU	NRZ	3.125 Gb/s	Ethernet	≥
FC133	NRZ	132.8 Mb/s	Fibre Channel	All
FC266	NRZ	265.6 Mb/s	Fibre Channel	All
FC531	NRZ	531.2 Mb/s	Fibre Channel	≥1.0 GHz
FC1063	NRZ	1.0625 Gb/s	Fibre Channel	
FC2125	NRZ	2.125 Gb/s	Fibre Channel	
FC4250	NRZ	4.256 Gb/s	Fibre Channel	
FST	NRZ	12.0 Mb/s	USB 1.1/2.0	All
FW1394 BS400B	NRZ	491.5 Mb/s	IEEE1394b	≥1.0 GHz
FW1394 BS1600B	NRZ	1.966 Gb/s	IEEE1394b	
HST	NRZ	480 Mb/s	USB 1.1/2.0	≥1.0 GHz
INF_5G	NRZ	2.5 Gb/s	Infiniband	
OC1	NRZ	51.84 Mb/s	SONET/SDH	All
OC3	NRZ	155.52 Mb/s	SONET/SDH	All
OC12	NRZ	622.08 Mb/s	SONET/SDH	≥1.0 GHz
OC48	NRZ	2.4883 Gb/s	Sonet	
OC48 FEC	NRZ	2.666 Gb/s	Sonet	
PCIEXPRESS	NRZ	2.5 Gb/s	PCI-Express	
RATE32Mbit	NRZ	32.064 Mb/s	ITU	
RATE97Mbit	NRZ	97.728 Mb/s	ITU	
RIO_500M	NRZ	500 Mb/s	RapidIO/LP-LVDS	≥1.0 GHz
RIO_750_M	NRZ	1 Gb/s	RapidIO/LP-LVDS	
RIO_1G	NRZ	1 Gb/s	RapidIO/LP-LVDS	
RIO_1_5_G	NRZ	2 Gb/s	RapidIO/LP-LVDS	
RIO_SERIAL_1G	NRZ	1.25 Gb/s	RapidIO/LP-LVDS	
RIO_SERIAL_2G	NRZ	2.5 Gb/s	RapidIO/LP-LVDS	
RIO_SERIAL_3G	NRZ	3.125 Gb/s	RapidIO/LP-LVDS	
SAS1_5	NRZ		SAS	
SAS3_?	NRZ		SAS	

**Table 2-46: Communication Trigger Standards (cont.)**

Argument	Code	Bit Rate	Mask Type	Models
SFI5_2	NRZ	2.488 Gb/s	OIF	
SFI5_3	NRZ	3.125 Gb/s	OIF	
STM0_CMI	NRZ	51.84 Mb/s	ITU	
STM0_HDBX	NRZ	51.84 Mb/s	ITU	
STM1E	NRZ	155.52 Mb/s	ITU	
STS1	NRZ	51.840 Mb/s	ANSI T1.102	
STS3	NRZ	155.52 Mb/s	ANSI T1.102	
TFI5_2	NRZ	2.488 Gb/s	OIF	
TFI5_3	NRZ	3.1104 Gb/s	OIF	
VIDEO270	NRZ	270 Mb/s	Video	≥500 MHz
VIDEO292M	NRZ	1.485 Gb/s	Video	≥2.5 GHz
VIDEO360	NRZ	360 Mb/s	Video	≥500 MHz
VSROC192	NRZ	1.24116 Gb/s	OIF	

**Examples** TRIGGER:A:COMMUNICATION:STANDARD E4 sets the standard to E4.

TRIGGER:A:COMMUNICATION:STANDARD? might return  
:TRIGGER:A:COMMUNICATION:SOURCE:STANDARD DS1.

## TRIGger:A:HOLDoff? (Query Only)

This query-only command returns the A trigger holdoff parameters. These parameters specify the time period during which the trigger circuitry is not looking to generate a trigger event. This command is equivalent to selecting Holdoff from the Trig menu and then viewing the current settings.

**Group** Trigger

**Syntax** TRIGger:A:HOLDoff?

**Related Commands** [TRIGger:A:HOLDoff:ACTUal?](#), [TRIGger:A:HOLDoff:BY](#), [TRIGger:A:HOLDoff:TIME](#)

**Examples** TRIGGER:A:HOLDoff? might return :TRIGGER:A:HOLDoff:TIME  
900.0000E-09;BY DEFAULT, indicating that the A edge trigger holdoff time  
(by default) is set to 900 ns.

## TRIGger:A:HOLDoff:ACTUal? (Query Only)

This query-only command returns the holdoff time actually used (expressed in seconds) by the A trigger. This command is equivalent to selecting Holdoff from the Trig menu and then viewing the current Trig Holdoff value.

**Group** Trigger

**Syntax** TRIGger:A:HOLDoff:ACTUal?

**Related Commands** [TRIGger:A:HOLDoff?](#), [TRIGger:A:HOLDoff:BY](#), [TRIGger:A:HOLDoff:TIME](#)

**Examples** TRIGGER:A:HOLDOFF:ACTUAL? might return :TRIGGER:A:HOLDOFF:ACTUAL 4.0000E-06, indicating that the holdoff time is set to 4  $\mu$ s.

## TRIGger:A:HOLDoff:BY

This command sets or queries the type of holdoff for the A trigger. Holdoff types are expressed as either user-specified time (TIME) or by an internally calculated minimum time value (DEFAULT/AUTO). This command is equivalent to selecting Holdoff from the Trig menu and then setting the Holdoff type.

**Group** Trigger

**Syntax** TRIGger:A:HOLDoff:BY {TIME|DEFAULT|RANDOM|AUTO}  
TRIGger:A:HOLDoff:BY?

**Related Commands** [TRIGger:A:HOLDoff:TIME](#)

**Arguments** TIME enables you to set the holdoff time via the [TRIGger:A:HOLDoff:TIME](#) command.

DEFAULT automatically calculates a holdoff time to use. This time is typically equivalent to the greater of 1/2 screen (5 divisions) of time or 250 ns. The maximum value is 12 s. For example, if the instrument is set to 1 ms/division then the default holdoff will be 1 ms/division x 25 divisions = 25 ms.

RANDOM

AUTO

**Examples** TRIGGER:A:HOLDOFF:BY TIME sets the holdoff to the “by time” setting. This enables you to set the holdoff time.

TRIGGER:A:HOLDOFF:BY? might return :TRIGGER:A:HOLDOFF:BY TIME, indicating that you will set the holdoff time.

## TRIGger:A:HOLDoff:TIME

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Holdoff from the Trig menu and then choosing the desired Trig Holdoff.

**Group** Trigger

**Syntax** TRIGger:A:HOLDoff:TIME <NR3>  
TRIGger:A:HOLDoff:TIME?

**Related Commands** [TRIGger:A:HOLDoff:BY](#)

**Arguments** <NR3> specifies the holdoff time in seconds. The range is from 250 ns through 12.0 s.

**Examples** TRIGGER:A:HOLDOFF:TIME10 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  $\mu$ s.

## TRIGger:A:I2C:ADDRESS:MODE

DPO7000 Series only: This command sets or queries the I2C address mode to 7 or 10 bit.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:ADDRESS:MODE {ADDR7|ADDR10}  
TRIGger:A:I2C:ADDRESS:MODE?

**Arguments** ADDR7 specifies the 7-bit I2C address mode.

ADDR10 specifies the 10-bit I2C address mode.

**Examples** TRIGGER:A:I2C:ADDRESS:MODE ADDR10 sets the I2C address mode to 10 bit.  
 TRIGGER:A:I2C:ADDRESS:MODE? might return  
 :TRIGGER:A:I2C:ADDRESS:MODE ADD7, indicating that the I2C address mode is set to 7 bit.

## TRIGger:A:I2C:ADDRESS:RWINClude

Sets the I2C read/write bit to address type.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:ADDRESS:RWINClude {OFF|ON}  
 TRIGger:A:I2C:ADDRESS:RWINClude?

**Arguments** ON specifies the I2C read/write bit to address type.  
 OFF specifies to stop the I2C read/write bit to address type.

**Examples** TRIGGER:A:I2C:ADDRESS:RWINCLUDE ON sets the read/write bit to address type.  
 TRIGGER:A:I2C:ADDRESS:RWINCLUDE? might return  
 :TRIGGER:A:I2C:ADDRESS:RWINCLUDE OFF, indicating that the read/write bit stops specifying the address type.

## TRIGger:A:I2C:ADDRESS:TYPE

DPO7000 Series only: This command sets or queries the I2C address type.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:ADDRESS:TYPE  
 {GENERALCALL|STARTbyte|TENbit|NONE}  
 TRIGger:A:I2C:ADDRESS:TYPE?

**Arguments** GENERALCALL specifies a generalcall address type.  
 STARTBYTE specifies a startbyte address type.  
 TENBIT specifies a tenbit address type.  
 NONE specifies I2C address type of none.

**Examples** TRIGGER:A:I2C:ADDRESS:TYPE NONE sets the I2C address type to none.  
TRIGGER:A:I2C:ADDRESS:TYPE? might return  
:TRIGGER:A:I2C:ADDRESS:TYPE TENBIT, indicating that the I2C address type is set to tenbit.

## TRIGger:A:I2C:ADDResS:VALue

DPO7000 Series only: This command sets or queries the binary address string used for the I2C trigger if the trigger condition is ADDR or ADDRANDDATA.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:ADDResS:VALue <QString>  
TRIGger:A:I2C:ADDResS:VALue?

**Arguments** <QString> is up to 7 bits or 10 bits depending on the address mode that specifies the address.

**Examples** TRIGGER:A:I2C:ADDRESS:VALUE "1011" sets the I2C address value to 1011.  
TRIGGER:A:I2C:ADDRESS:VALUE? might return  
:TRIGGER:A:I2C:ADDRESS:VALUE "10111", indicating that the I2C address value is set to 10111.

## TRIGger:A:I2C:CLOCK:LEVel

DPO7000 Series only: This command sets or queries the clock source for the I2C Serial Trigger.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:CLOCK:LEVel <NR3>  
TRIGger:A:I2C:CLOCK:LEVel?

**Arguments** <NR3> specifies the I2C clock level. The clock level range is  $\pm 5$ .

**Examples** TRIGGER:A:I2C:CLOCK:LEVEL0.0 sets the I2C clock level to 0.0.  
 TRIGGER:A:I2C:CLOCK:LEVEL? might return  
 :TRIGGER:A:I2C:CLOCK:LEVEL 5.0000, indicating that the I2C clock level  
 is set to 5.

## TRIGger:A:I2C:CLOCK:SOURce

DPO7000 Series only: This command sets or queries the clock source for the I2C Serial Trigger.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:CLOCK:SOURCE CH<x>  
 TRIGger:A:I2C:CLOCK:SOURCE?

**Arguments** CH<x> specifies the clock source. x can be 1 through 4.

**Examples** TRIGGER:A:I2C:CLOCK:SOURCE CH1 sets the I2C clock source to channel CH1.  
 TRIGGER:A:I2C:CLOCK:SOURCE? might return  
 :TRIGGER:A:I2C:CLOCK:SOURCE CH3, indicating that the I2C clock source  
 is set to CH3.

## TRIGger:A:I2C:CONDition

DPO7000 Series only: This command sets or queries the trigger condition for the I2C trigger.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:CONDition  
 {START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}  
 TRIGger:A:I2C:CONDition?

**Arguments** START specifies triggering on the start of a packet.  
 STOP specifies triggering on the stop packet.  
 REPEATstart specifies triggering on a repeated start of packet.  
 ACKMISS specifies triggering on a missing acknowledgement.

ADDRESS specifies triggering on the address.

DATA specifies triggering on the specified data.

ADDRANDDATA specifies triggering on the specified data and address.

- Examples** TRIGGER:A:I2C:CONDITION START sets the trigger condition for the I2C trigger on the start of the packet.
- TRIGGER:A:I2C:CONDITION? might return :TRIGGER:A:I2C:CONDITION DATA, indicating that the trigger condition for the I2C trigger is set on specified data.

## TRIGger:A:I2C:DATA:DIRection

DPO7000 Series only: This command sets or queries the I2C trigger condition valid on a READ, WRITE, or either.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:DATA:DIRection {READ|WRITE|NOCARE}  
TRIGger:A: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:I2C:DATA:DIRECTION WRITE sets the I2C data direction to write.
- TRIGGER:A:I2C:DATA:DIRECTION? might return :TRIGGER:A:I2C:DATA:DIRECTION READ, indicating that the I2C data direction is set on read.

## TRIGger:A:I2C:DATA:LEVel

DPO7000 Series only: This command sets or queries the threshold level for the I2C data source.

**Group** Low Speed Serial Trigger



<b>Syntax</b>	TRIGger:A:I2C:DATA:LEVEL <NR3> TRIGger:A:I2C:DATA:LEVEL?
<b>Arguments</b>	<NR3> specifies the I2C data level with a range of $\pm\%.0$ .
<b>Examples</b>	TRIGGER:A:I2C:DATA:LEVEL 0.0 sets the I2C data level to 0.0.  TRIGGER:A:I2C:DATA:LEVEL? might return :TRIGGER:A:I2C:DATA:LEVEL 0.0, indicating that the I2C data level is set on 0.0.

## TRIGger:A:I2C:DATA:SOURce

DPO7000 Series only: This command sets or queries the data source for the I2C serial trigger.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:I2C:DATA:SOURce CH<x> TRIGger:A:I2C:DATA:SOURce?
<b>Arguments</b>	CH<x> specifies the data source. x can be 1 through 4.
<b>Examples</b>	TRIGGER:A:I2C:DATA:SOURCE CH1 sets the data source for the I2C serial trigger to channel CH1.  TRIGGER:A:I2C:DATA:SOURCE? might return :TRIGGER:A:I2C:DATA:SOURCE CH4, indicating that the data source for the I2C serial trigger is set on channel CH4.

## TRIGger:A:I2C:DATA:START

DPO7000 Series only: This command sets or queries the data start byte. This byte is only needed if triggering on data. This byte is the first byte that pattern matching uses. The byte is a destination offset from the start of the packet.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:I2C:DATA:START <NR3> TRIGger:A:I2C:DATA:START?

**Arguments** <NR3> specifies the data start byte (offset), with a range of 0 to 65535.

**Examples** TRIGGER:A:I2C:DATA:START 0 specifies the I2C data start byte offset to 0.  
TRIGGER:A:I2C:DATA:START? might return :TRIGGER:A:I2C:DATA:START 5, indicating that the I2C data start byte offset is set to 5.

## TRIGger:A:I2C:DATA:VALue

DPO7000 Series only: This command sets or queries the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:DATA:VALue <QString>  
TRIGger:A:I2C:DATA:VALue?

**Arguments** <QString> sets the I2C data value. This value can be 1 to 32 bits.

**Examples** TRIGGER:A:I2C:DATA:VALUE "1011" sets the I2C data value to 1011.  
TRIGGER:A:I2C:DATA:VALUE? might return :TRIGGER:A:I2C:DATA:VALUE "10111", indicating that the I2C data value is set to 10111.

## TRIGger:A:I2C:FORMat

DPO7000 Series only: This command sets or queries the display format for the I2C data value.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:I2C:FORMat {BINary|HEX}  
TRIGger:A:I2C:FORMat?

**Arguments** BINary specifies binary format for I2C triggering.  
HEX specifies hexadecimal format for I2C triggering.

**Examples** TRIGGER:A:I2C:FORMAT HEX sets the I2C trigger format to hexadecimal.

TRIGGER:A:I2C:FORMAT? might return :TRIGGER:A:I2C:FORMAT BINARY, indicating that the I2C trigger format is set to binary.

## TRIGger:A:MODE

This command sets or queries the A trigger mode. This command is equivalent to selecting Mode from the Trig menu and then choosing the desired Trigger Mode.

**Group** Trigger

**Syntax** TRIGger:A:MODE {AUTO|NORMa1}  
TRIGger:A:MODE?

**Related Commands** [TRIGger:{A|B}:LEVel](#)

**Arguments** AUTO generates a trigger if one is not detected within a specified time period.  
NORMa1 waits for a valid trigger event.

**Examples** TRIGGER:A:MODENORMAL sets the trigger mode to normal.

TRIGGER:A:MODE? might return :TRIGGER:A:MODE AUTO, indicating that the trigger mode is auto.

## TRIGger:A:RS232:BAUd

DPO7000 Series only: This command sets or queries the baud rate for RS232 triggering.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:RS232:BAUd <x>  
TRIGger:A:RS232:BAUd?

**Arguments** <x> specifies the baud rate for RS232 triggering. X can be: 1500000, 921600, 750000, 460800, 115200, 57600, 38400, 19200, 9600, 7200, 4800, 2400, 2000, 1800, 1200, 600, 300, 150, or 75.

**Examples** TRIGGER:A:RS232:BAUD 2000 sets the RS232 triggering baud rate to 2000.  
TRIGGER:A:RS232:BAUD? might return :TRIGGER:A:RS232:BAUD 1800, indicating that the RS232 triggering baud rate is set to 1800.

## TRIGger:A:RS232:DATA:LEVEL

DPO7000 Series only: This command sets or queries the threshold for the RS232 data source.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:RS232:DATA:LEVEL <NR3>  
TRIGger:A:RS232:DATA:LEVEL?

**Arguments** <NR3> specifies the threshold level with a range of  $\pm 5.0$ .

**Examples** TRIGGER:A:RS232:DATA:LEVEL 0.0 sets the threshold for the RS232 data source to 0.0.  
TRIGGER:A:RS232:DATA:LEVEL? might return :TRIGGER:A:RS232:DATA:LEVEL 5.0, indicating that the threshold for the RS232 data source is set to 5.0.

## TRIGger:A:RS232:DATA:SOURCE

DPO7000 Series only: This command sets or queries the RS232 data source.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:RS232:DATA:SOURCE CH<x>  
TRIGger:A:RS232:DATA:SOURCE?

**Arguments** CH<x> specifies the RS232 data source. x can be 1, 2, 3, or 4.

**Examples** TRIGGER:A:RS232:DATA:SOURCE CH2 sets the RS232 data source to channel CH2.  
TRIGGER:A:RS232:DATA:SOURCE? might return :TRIGGER:A:RS232:DATA:SOURCE CH1, indicating that the RS232 data source is set to channel CH1.

## TRIGger:A:RS232:DATA:VALue

DPO7000 Series only: This command sets or queries the binary data string used for RS232 triggering.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:RS232:DATA:VALue <QString> TRIGger:A:RS232:DATA:VALue?
<b>Arguments</b>	<QString> specifies the RS232 data value. The value can be up to 8 bits.
<b>Examples</b>	TRIGGER:A:RS232:DATA:VALUE "1011" sets the RS232 triggering data value to 1011.  TRIGGER:A:RS232:DATA:VALUE? might return :TRIGGER:A:RS232:DATA:VALUE "10111", indicating that the RS232 data value is set to 10111.

## TRIGger:A:RS232:FORMat

DPO7000 Series only: This command sets or queries the display format for the RS232 data value.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:RS232:FORMat {BINary HEX} TRIGger:A:RS232:FORMat?
<b>Arguments</b>	BINary specifies binary for the display format.  HEX specifies hexadecimal for the display format.
<b>Examples</b>	TRIGGER:A:RS232:FORMAT HEX sets the display format for the RS232 data to hexadecimal.  TRIGGER:A:RS232:FORMAT? might return :TRIGGER:A:RS232:DATA:FORMAT BINARY, indicating that the display format for the RS232 data is set to binary.

## TRIGger:A:RS232:PARity

DPO7000 Series only: This command sets or queries the state of the RS232 parity bit.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:RS232:PARity {NONE|EVEN|ODD}  
TRIGger:A:RS232:PARity?

**Arguments** NONE specifies no parity.  
EVEN specifies even RS232 parity.  
ODD specifies odd RS232 parity.

**Examples** TRIGGER:A:RS232:PARITYEVEN sets the RS232 parity to even.  
TRIGGER:A:RS232:PARITY? might return :TRIGGER:A:RS232:PARITY ODD, indicating that the RS232 parity is set to odd.

## TRIGger:A:SERIAL:BITRate

This command sets or queries the clock/data bit rate. Changing the bit rate causes the standard to become “custom”. It remains “custom” until another standard is chosen.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:BITRate <NR3>  
TRIGger:A:SERIAL:BITRate?

**Arguments** <NR3> This is the series bit rate and is expressed in bits per second. The range is 1.5e6 to 1.25e9.

**Examples** TRIGGER:A:SERIAL:BITRATE 51.8400E+6 sets the bit rate to 51.84 Mbits/second.  
TRIGGER:A:SERIAL:BITRATE? might return :TRIGGER:A:SERIAL:BITRATE 51.8400E+6, indicating that the bit rate is set to 51.84 Mbits/second.

## TRIGger:A:SERIAL:CLOCK:LEVEL

This command sets or queries the serial trigger clock level.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:CLOCK:LEVEL <NR3>  
TRIGger:A:SERIAL:CLOCK:LEVEL?

**Arguments** <NR3> is a value in the range of 9.9E37 and is expressed in volts.

**Examples** TRIGGER:A:SERIAL:CLOCK:LEVEL 0.0 sets the clock level to 0 V.  
TRIGGER:A:SERIAL:CLOCK:LEVEL? might return  
:TRIGGER:A:SERIAL:CLOCK:LEVEL 51.8400E+6, indicating that the  
32.0000E-3 indicating that the serial trigger clock level is set to 51.84E6 volts.

## TRIGger:A:SERIAL:CLOCK:POLarity

This command sets or queries the serial clock polarity.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:CLOCK:POLarity {RISe|FALL}  
TRIGger:A:SERIAL:CLOCK:POLarity?

**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.

**Examples** TRIGGER:A:SERIAL:CLOCK:POLARITY RISE sets the clock polarity to rise.  
TRIGGER:A:SERIAL:CLOCK:POLARITY? might return  
:TRIGGER:A:SERIAL:CLOCK:POLARITY FALL, indicating that the  
clock polarity is set to fall.



## TRIGger:A:SERIAL:CLOCK:SOURce

This command sets or queries the serial data source channel.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SERIAL:CLOCK:SOURce {CH<x> RECOVERed} TRIGger:A:SERIAL:CLOCK:SOURce?
<b>Arguments</b>	CH<x> specifies one of the input channels. x can be 1, 2, 3, or 4.  RECOVERed specifies clock polarity and level are extracted from the serial data stream and other clock parameters are ignored.
<b>Examples</b>	TRIGGER:A:SERIAL:CLOCK:SOURCE CH1 sets the serial source channel to CH1.  TRIGGER:A:SERIAL:CLOCK:SOURCE? might return :TRIGGER:A:SERIAL:CLOCK:SOURCE CH3, indicating that the serial source channel is set to CH3.

## TRIGger:A:SERIAL:CODE

This command sets or queries the signal code.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SERIAL:CODE {NRZ S8B10B} TRIGger:A:SERIAL:CODE?
<b>Arguments</b>	NRZ sets the code to NRZ.  S8B10B sets the code to S8B0B.
<b>Examples</b>	TRIGGER:A:SERIAL:CODE NRZ sets the signal code to NRZ.  TRIGGER:A:SERIAL:CODE? might return :TRIGGER:A:SERIAL:CODE S8B10B, indicating that the signal code is set to S8B0B.

## TRIGger:A:SERIAL:DATA:FORMAT

This command sets or queries how the Pattern string is formatted.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:DATA:FORMAT {BINary|HEX}  
TRIGger:A:SERIAL:DATA:FORMAT?

**Arguments** BINary pattern symbols are: {'SPACE'|0|1|X} 'SPACE' is white space and can be included to make the 32-bit pattern easier to read when setting the pattern.

HEX pattern symbols are used only with NRZ. They are: {'|0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F|X|?'} Hex 'X' represents the bit pattern 'XXXX'. Hex '?' represents any other binary bit pattern which doesn't have a hex representation, such as '00X1'.

**Examples** TRIGGER:A:SERIAL:DATA:FORMAT BINARY sets the Pattern string to a binary format.

TRIGGER:A:SERIAL:DATA:FORMAT? might return  
:TRIGGER:A:SERIAL:CODE HEX, indicating that the Pattern string is set to a hexadecimal format.

## TRIGger:A:SERIAL:DATA:PATtern

This command sets or queries the data pattern to allow up to 32-bit serial patterns.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:DATA:PATtern <QString>  
TRIGger:A:SERIAL:DATA:PATtern?

**Related Commands** [TRIGger:A:SERIAL:DATA:PATtern:NRZ](#)  
[TRIGger:A:SERIAL:DATA:PATtern:S8B10B](#)

**Arguments** <QString> specifies the serial pattern to trigger on. The default value is "01", such that there are up to 64 characters total.

**Examples** If one sets the data format to binary with a `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN?` might return the following characters: `1101XXXX10` :

If one next sets the data format to hexadecimal with the `TRIGGER:A:SERIAL:DATA:FORMAT` commands as follows: `TRIGGER:A:SERIAL:DATA:FORMAT HEX` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN?` might return the following characters: `3??` Showing, in this case, 3 followed by two “?”. Each hex character has replaced four binary characters. The hex “??” has replaced the binary “01XXXX10”. The “?” was used because “01XX” and “XX10” don’t have an exact hex representations.

If one sends: `TRIGGER:A:SERIAL:DATA:PATTERN “XXXX XXXa”` this will set the pattern to eight hex digits. The space in the middle only serves to make the pattern easier to read.

Now, if one sends the following query: `TRIGGER:A:SERIAL:DATA:PATTERN?` this might return: `XXXXXXXXA` showing the effect of the pattern sent in example 3. The cosmetic space from example 3 is not returned.

Finally, one could change the data format back to binary with the `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then one could query the instrument using the pattern command, as follows: `TRIGGER:A:SERIAL:DATA:PATTERN?` This could return: `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1010` showing the response due to the binary format. Four binary characters “1010” now replace the hex character “A”.

## TRIGger:A:SERIAL:DATA:PaTtern:NRZ

This command sets or queries the data pattern to allow up to 32-bit serial patterns.

**Group** Low Speed Serial Trigger

**Syntax** `TRIGger:A:SERIAL:DATA:PaTtern:NRZ <QString>`  
`TRIGger:A:SERIAL:DATA:PaTtern:NRZ?`

**Related Commands** [TRIGger:A:SERIAL:DATA:PaTtern](#)  
[TRIGger:A:SERIAL:DATA:PaTtern:S8B10B](#)

**Arguments** `<QString>` specifies the serial pattern to trigger on. The default value is “01”, such that there are up to 64 characters total.

**Examples** If one sets the data format to binary with a `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:NRZ?` might return the following characters: `1101XXXX10` :

If one next sets the data format to hexadecimal with the `TRIGGER:A:SERIAL:DATA:FORMAT` commands as follows: `TRIGGER:A:SERIAL:DATA:FORMAT HEX` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:NRZ?` might return the following characters: `3??` Showing, in this case, 3 followed by two “?”. Each hex character has replaced four binary characters. The hex “??” has replaced the binary “01XXXX10”. The “?” was used because “01XX” and “XX10” don’t have an exact hex representations.

If one sends: `TRIGGER:A:SERIAL:DATA:PATTERN:NRZ “XXXX XXXa”` this will set the pattern to eight hex digits. The space in the middle only serves to make the pattern easier to read.

Now, if one sends the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:NRZ?` this might return: `XXXXXXXXA` showing the effect of the pattern sent in example 3. The cosmetic space from example 3 is not returned.

Finally, one could change the data format back to binary with the `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then one could query the instrument using the pattern command, as follows: `TRIGGER:A:SERIAL:DATA:PATTERN:NRZ?` This could return: `XXXXXXXXXXXXXXXXXXXXXXXXXXXX1010` showing the response due to the binary format. Four binary characters “1010” now replace the hex character “A”.

## TRIGger:A:SERIAL:DATA:PaTtern:S8B10B

This command sets or queries the data pattern to allow up to 32-bit serial patterns.

**Group** Low Speed Serial Trigger

**Syntax** `TRIGger:A:SERIAL:DATA:PaTtern:S8B10B <QString>`  
`TRIGger:A:SERIAL:DATA:PaTtern:S8B10B?`

**Related Commands** [TRIGger:A:SERIAL:DATA:PaTtern](#)  
[TRIGger:A:SERIAL:DATA:PaTtern:NRZ](#)

**Arguments** `<QString>` specifies the serial pattern to trigger on. The default value is “0011111010”, such that there are up to 64 characters total.

**Examples** If one sets the data format to binary with a `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:S8B10B?` might return the following characters: `1101XXXX10` :

If one next sets the data format to hexadecimal with the `TRIGGER:A:SERIAL:DATA:FORMAT` commands as follows: `TRIGGER:A:SERIAL:DATA:FORMAT HEX` then the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:S8B10B?` might return the following characters: `3??` Showing, in this case, 3 followed by two “?”. Each hex character has replaced four binary characters. The hex “??” has replaced the binary “`01XXXX10`”. The “?” was used because “`01XX`” and “`XX10`” don’t have an exact hex representations.

If one sends: `TRIGGER:A:SERIAL:DATA:PATTERN:S8B10B "XXXX XXXa"` this will set the pattern to eight hex digits. The space in the middle only serves to make the pattern easier to read.

Now, if one sends the following query: `TRIGGER:A:SERIAL:DATA:PATTERN:S8B10B?` this might return: `XXXXXXXXA` showing the effect of the pattern sent in example 3. The cosmetic space from example 3 is not returned.

Finally, one could change the data format back to binary with the `TRIGGER:A:SERIAL:DATA:FORMAT` command, as follows: `TRIGGER:A:SERIAL:DATA:FORMAT BIN` then one could query the instrument using the pattern command, as follows: `TRIGGER:A:SERIAL:DATA:PATTERN:S8B10B?` This could return: `XXXXXXXXXXXXXXXXXXXXXXXXXXXX1010` showing the response due to the binary format. Four binary characters “`1010`” now replace the hex character “`A`”.

## TRIGGER:A:SERIAL:ERRORdetector:ALIGNCHARacter? (Query Only)

This command queries both of the align character values. Align characters must be defined for all test types, and those characters must appear in the signal test pattern.

**Group** Error Detector

**Syntax** `TRIGGER:A:SERIAL:ERRORdetector:ALIGNCHARacter?`

**Examples** `TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER?` might return `:TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDPLUS "1100000101";RDMINUS "0011111010"`, indicating that the RDPLUS align character is `1100000101` and the RDMINUS align character is `0011111010`.

## TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDMINUS

This command sets or queries the RD- align character value.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDMINUS <Qstring> TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDMINUS?
<b>Arguments</b>	<Qstring> is a quoted string representing a 10-bit character value.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDMINUS "0011111010" sets the RD- align character value to "0011111010".  TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDMINUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDMINUS "0011111010", indicating that the RD- align character value is "0011111010". Reception of this character causes the receiver to align to the 10-bit character boundary.

## TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDPLUS

This command sets or queries the RD+ align character value.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDPLUS <Qstring> TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDPLUS?
<b>Arguments</b>	<Qstring> is a quoted string representing a 10-bit character value.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDPLUS "1100000101" sets that the RD+ align character to "1100000101".  TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDPLUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNCHARACTER:RDPLUS "1100000101", indicating that the RD+ align character value is 1100000101. Reception of the align character aligns the receiver to the 10-bit character boundary.

## TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive? (Query Only)

This command queries all of the align primitive values.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE? might return  
 :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:S8BPATTERN  
 "00111101010100100101001011011110";RDPLUS:ENABLE  
 OFF;PATTERN "110000010101010101010101010101011101100011";  
 :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS:ENABLE  
 OFF;PATTERN "0011111010010101010", indicating that the S8B pattern is  
 00111101010100100101001011011110, RDPLUS is enabled, the RDPLUS  
 pattern is 110000010101010101010101010101011101100011, RDMINUS is  
 enabled, and the RDMINUS pattern is 0011111010010101010.

## TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS? (Query Only)

This command queries the RD- align primitive enable and string values.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS? might  
 return :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS  
 ON,"110000010101010101010101010101011101100011", indicating  
 that the RD- align primitive option is enabled and the pattern is  
 "110000010101010101010101010101011101100011".

## TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS:ENABLE

This command sets or queries the status of the RD- align primitive option.

**Group** Error Detector

**Syntax** TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDMINUS:ENABLE  
{ON|OFF}  
TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDMINUS:  
ENABLE?

**Arguments** ON enables the RD- align primitive option.  
OFF disables the RD- align primitive option.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS:ENABLE  
ON enables the RD- align primitive option.  
  
TRIGGER:A:SERIAL:ERRORDETECTOR:  
ALIGNPRIMITIVE:RDMINUS:ENABLE? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS:ENABLE  
ON, indicating that the RD- align primitive option is enabled.

## TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDMINUS:PATTERN

This command sets or queries the RD- align primitive pattern.

**Group** Error Detector

**Syntax** TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDMINUS:  
PATTERN <Qstring>  
TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDMINUS:  
PATTERN?

**Arguments** <Qstring> is a quoted string representing a 40-bit RD- align primitive pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS:  
PATTERN "00111110100101010101010101010101010010011100" sets that the  
RD- align primitive pattern to "00111110100101010101010101010101010010011100".  
  
TRIGGER:A:SERIAL:ERRORDETECTOR:  
ALIGNPRIMITIVE:RDMINUS:PATTERN? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDMINUS:PATTERN  
"00111110100101010101010101010101010010011100", indicating that the RD-  
align primitive pattern is "00111110100101010101010101010101010010011100".

## TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDPLUS? (Query Only)

This command queries the RD+ align primitive enable and string values.



<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: ALIGNPRIMITIVE:RDPLUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:ENABLE ON, "110000010101010101010101010101011101100011", indicating that the RD+ align primitive option is enabled and the pattern is "110000010101010101010101010101011101100011".

## TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:ENABLE

This command sets or queries the status of the RD+ align primitive option.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:ENABLE <ON OFF> TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:ENABLE?
<b>Arguments</b>	ON enables the RD+ align primitive option. OFF disables the RD+ align primitive option.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:ENABLE ON, enables the RD+ align primitive option.  TRIGGER:A:SERIAL:ERRORDETECTOR: ALIGNPRIMITIVE:RDPLUS:ENABLE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:ENABLE ON, indicating that the RD+ align primitive option is enabled.

## TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:PATtern

This command sets or queries the RD+ align primitive pattern.

<b>Group</b>	Error Detector
--------------	----------------

**Syntax** TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDPLUS:PATTERN  
<Qstring>  
TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:RDPLUS:  
PATTERN?

**Arguments** <Qstring> is a quoted string representing a 40-bit RD+ align primitive pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:PATTERN  
"110000010101010101010101010101011101100011" sets the RD+ align  
primitive pattern to "110000010101010101010101010101011101100011".

TRIGGER:A:SERIAL:ERRORDETECTOR:  
ALIGNPRIMITIVE:RDPLUS:PATTERN? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:PATTERN  
"110000010101010101010101010101011101100011",  
indicating that the RD+ align primitive pattern is  
"110000010101010101010101010101011101100011".

## TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:S8BPATTERN

This command sets or queries the 32-bit version for the align primitive pattern.

**Group** Error Detector

**Syntax** TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:S8BPATTERN  
<Qstring>  
TRIGGER:A:SERIAL:ERRORdetector:ALIGNPRIMITIVE:S8BPATTERN?

**Arguments** <Qstring> is a quoted string representing a 32-bit align primitive pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:S8BPATTERN  
"00111101010100100100101001011011110" sets the align primitive pattern to  
"00111101010100100100101001011011110".

TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:S8BPATTERN?  
might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:ALIGNPRIMITIVE:RDPLUS:PATTERN  
"00111101010100100100101001011011110", indicating that the align primitive  
pattern is "00111101010100100100101001011011110".

## TRIGger:A:SERIAL:ERRORdetector:BIT:LENGTH

This command sets or queries the signal test pattern length in bits for non-USB bit error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:BIT:LENGTH <NR1> TRIGger:A:SERIAL:ERRORdetector:BIT:LENGTH?
<b>Arguments</b>	<NR1> indicates the bit length of the signal test pattern.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:LENGTH 800 sets the signal test pattern length to 800.  TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:LENGTH? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:LENGTH 800, indicating that the signal test pattern length is 800.

## TRIGger:A:SERIAL:ERRORdetector:BIT:SYNCPATtern

This command sets or queries the bit sync pattern for nonUSB bit error testing. The SYNCPATtern consists of one to four 10-bit 8B10B characters. It is required for all non-USB bit error testing. This sequence of 8B10B characters must actually occur in the signal test pattern sent to the oscilloscope.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:BIT:SYNCPATtern <Qstring> TRIGger:A:SERIAL:ERRORdetector:BIT:SYNCPATtern?
<b>Arguments</b>	<Qstring> is a quoted string representing a 10bit pattern. The default is K28.5, "0011111010", and "110000101".
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:SYNCPATTERN "1101001010" sets the sync pattern to "1101001010".  TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:SYNCPATTERN? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:SYNCPATTERN "1101001010", indicating that the sync pattern is "1101001010".

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST (No Query Form)

This command initiates and terminates bit error testing for the arguments START and STOP. It zeroes bit error test results for the argument CLEAR . Also copies the test pattern from the signal for the argument LEARN.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST  
<LEARN|START|STOP|CLEAR>

**Arguments** LEARN copies the test pattern from the signal to the hardware template memory.  
START initiates bit error counting of bit errors and duration.  
STOP terminates bit error counting of bit errors and duration.  
CLEAR zeroes the bit error count, duration, and bit error rate.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST LEARN copies the test pattern of the signal to the hardware template memory.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:COUNT? (Query Only)

This command queries the bit error count for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:COUNT?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:COUNT? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:COUNT 10, indicating that the bit error count for bit error testing is 10.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:DURATION? (Query Only)

This command queries the elapsed duration (in units of bits) tested for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:DURATION?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:DURATION? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:DURATION 7612, indicating that the elapsed duration (in units of bits) for bit error testing is 7612.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:MAXALIGNs? (Query Only)

This command queries the maximum consecutive SATA align primitives or USB skip order sets for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:MAXALIGNs?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:MAXALIGNs? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:MAXALIGNs 3, indicating that the maximum consecutive SATA align primitives or USB skip order sets encountered for bit error testing is 3.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RATE? (Query Only)

This command queries the calculated bit error rate for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:RATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:RATE 0.0000, indicating that the calculated bit error rate for bit error testing is 0.0000.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RESULTS? (Query Only)

This command queries all of the results for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:RESULTS?

**Returns** COUNT which is the bit error count (number of bad bits).  
DURATION which is the elapsed duration (number of bits tested).  
RATE which is the calculated bit error rate.  
MAXALIGNS which is the maximum consecutive aligns encountered.  
SECONDS which is the elapsed time in seconds.  
TIME which is the elapsed duration time in Days, Hours, Minutes, and Seconds.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:RESULTS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:RESULTS 2,100000000000,0.2 e-12,3,15,0,0,0,15, indicating that the bit error count is 2, elapsed duration is 100000000000, calculated bit error rate is 0.2 e-12, maximum consecutive aligns encountered is 3, elapsed time is 15 seconds, and the elapsed time is 0 days, 0 hours, 0 minutes, and 15 seconds for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:SECONDS? (Query Only)

This command queries the elapsed time in seconds for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:SECONDS 30, indicating that the elapsed time is 30 seconds.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS? (Query Only)

This command queries all of the bit error test status bits.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS?

**Returns** COUNTING or NOT\_COUNTING for START status for bit error testing.  
 LOCKED or NOT\_LOCKED for LOCK status for bit error testing.  
 SYNC or NOT\_SYNC for SYNC status for bit error testing.  
 SIGNAL or NO\_SIGNAL for SIGNAL status for bit error testing.  
 MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for bit error testing.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS STOPPED;LOCKED;NOT\_SYNC;NO\_SIGNAL;MAX\_AP\_OK, indicating that the bits error status are STOPPED, LOCKED, NOT\_SYNC, NO\_SIGNAL, and MAX\_AP\_OK.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:MAX\_AP? (Query Only)

This command queries the MAX\_AP status for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:MAX\_AP?

**Returns** MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for bit error testing.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:MAX\_AP? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:MAX\_AP MAX\_AP\_OK, indicating that the status of MAX\_AP is ok.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:LOCK? (Query Only)

This command queries the LOCK status for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:LOCK?

**Returns** LOCKED or NOT\_LOCKED for LOCK status for bit error testing.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:LOCK? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:LOCK LOCKED, indicating that the recovered clock reference signal is LOCKED.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SIGNAL? (Query Only)

This command queries the SIGNAL status for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SIGNAL?

**Returns** SIGNAL or NO\_SIGNAL for SIGNAL status for bit error testing.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:SIGNAL? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:SIGNAL SIGNAL, indicating that the signal is connected for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:START? (Query Only)

This command queries the START status for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:START?

**Returns** COUNTING or NOT\_COUNTING for START status for bit error testing.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:START? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:START COUNTING, indicating that the bit error comparisons are being made.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SYNC? (Query Only)

This command queries the SYNC status for bit error testing.

**Group** Error Detector



<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:STATUS:SYNC?
<b>Returns</b>	SYNC or NOT_SYNC for SYNC status for bit error testing.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:SYNC? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:STATUS:SYNC NOT_SYNC, indicating the status of the SYNC for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME? (Query Only)

This command queries the elapsed time (in days, hours, minutes, and seconds) for bit error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME 0,0,3,14, indicating that the elapsed time is 0 days, 0 hours, 3 minutes, and 14 seconds for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:DAY? (Query Only)

This command queries the elapsed time days component for bit error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:DAY?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:DAY? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:DAY 0, indicating that the elapsed time days component is 0 for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:HOURS? (Query Only)

This command queries the elapsed time hours component for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:HOURS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:HOURS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:HOURS 0, indicating that the elapsed time hours component is 0 for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:MINUTES? (Query Only)

This command queries the elapsed time minutes component for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:MINUTES?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:MINUTES? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:MINUTES 0, indicating that the elapsed time minutes component is 0 for bit error testing.

## TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:SECONDS? (Query Only)

This command queries the elapsed time seconds component for bit error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:TIME:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:BIT:TEST:TIME:SECONDS 20, indicating that the elapsed time seconds component for bit error testing is 20.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER? (Query Only)

This command queries all character error settings, status, and results.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:START  
STOPPED;LOCK NOT\_LOCKED;SIGNAL NO\_SIGNAL;MAX\_AP  
MAX\_AP\_OK;;:TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME:DAY  
S 0;HOURS 0;MINUTES 3;SECONDS 0, indicating that the test is stopped, not  
locked, there is no signal, MAX\_AP is ok, and the test time is 3 minutes.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST (No Query Form)

This command initiates and terminates character error testing for the argument START and STOP. Also zeroes the character error results for the argument CLEAR.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST  
<START|STOP|CLEAR>

**Arguments** START initiates character error testing.  
STOP terminates character error testing.  
CLEAR zeroes the character error count, duration, and error rate.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST START initiates  
character error testing.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:COUNT? (Query Only)

This command queries the number of illegal characters in the signal for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:COUNT?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:COUNT? might  
return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:COUNT 0,

indicating that the number of illegal characters in the signal for character error testing is 0.

## **TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DISPCOUNT? (Query Only)**

This command queries the number of disparity errors in the signal for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DISPCOUNT?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DISPCOUNT? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DISPCOUNT 0, indicating that the number of disparity errors in the signal for character error testing is 0.

## **TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DISPRATE? (Query Only)**

This command queries the calculated disparity error rate for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DISPRATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DISPRATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DISPRATE 0.0000, indicating that the calculated disparity error rate for character error testing is 0.0000.

## **TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DURATION? (Query Only)**

This command queries the elapsed duration in units of characters tested.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:DURATION?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DURATION? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:DURATION 2876, indicating that the elapsed duration in units of characters tested is 2876.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:MAXALIGNs? (Query Only)

This command queries the maximum consecutive aligns encountered for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:MAXALIGNs?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:MAXALIGNs? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:MAXALIGNs 2, indicating that the maximum consecutive aligns encountered for character error testing is 2.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:RATE? (Query Only)

This command queries the calculated illegal character error rate for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:RATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:RATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:RATE 0.00000, indicating that the calculated illegal character error rate for character error testing is 0.0000.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:RESULTS? (Query Only)

This command queries all of the character error test results.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:RESULTS?

**Returns** COUNT is the number of illegal characters in the signal.  
DURATION is the elapsed duration in units of characters tested.  
RATE is the calculated illegal character error rate.  
MAXALIGNS is the maximum consecutive aligns encountered.  
DISPCOUNT is the number of disparity errors in the signal.  
DISPRATE is the calculated disparity error rate.  
SECONDS is the elapsed duration time in seconds.  
TIME is the elapsed duration time in DD:HH:MM:SS.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:RESULTS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:RESULTS 0,2365,0.0000,2,2,0.0000,16,0,0,2,16 , indicating that the number of illegal characters in the signal is 0, elapsed duration in units of characters tested is 2365, calculated illegal character error rate is 0.0000, maximum consecutive aligns encountered is 2, number of disparity errors in the signal is 2, calculated disparity error rate is 0.0000, elapsed duration time in seconds is 16, and the elapsed duration time is 0 days, 0 hours, 2 minutes, and 16 seconds.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:SECONDS? (Query Only)

This command queries the results of the elapsed time (in seconds) for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:SECONDS 30, indicating that elapsed time for character error testing is 30 seconds.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS? (Query Only)

This command queries all the character error test status.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS?

**Returns** COUNTING or NOT\_COUNTING for START status for character error test.  
 LOCKED or NOT\_LOCKED for LOCK status for character error test.  
 SIGNAL or NO\_SIGNAL for SIGNAL status for character error test.  
 MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for character error test. MAX\_AP.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS STOPPED;LOCKED;NO\_SIGNAL;MAX\_AP\_OK, indicating that the status for character error test are STOPPED, LOCKED, NO\_SIGNAL, and MAX\_AP\_OK.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:MAX\_AP? (Query Only)

This command queries the MAX\_AP status for the character error test.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:MAX\_AP?

**Returns** MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for character error test. MAX\_AP.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:MAX\_AP? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:MAX\_AP MAX\_AP\_OK, indicating that the MAX\_AP status is ok for character error test.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:LOCK? (Query Only)

This command queries the LOCK status for the character error test.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:LOCK?
<b>Returns</b>	LOCKED or NOT_LOCKED for LOCK status for character error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: CHARACTER:TEST:STATUS:LOCK? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:LOCK LOCKED, indicating that the recovered clock reference signal is LOCKED for error test.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:SIGNAL? (Query Only)

This command queries the SIGNAL status for the character error test.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:SIGNAL?
<b>Returns</b>	SIGNAL or NO_SIGNAL for SIGNAL status for character error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: CHARACTER:TEST:STATUS:SIGNAL? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:SIGNAL SIGNAL, indicating that the signal is connected for character error test.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:START? (Query Only)

This command queries the START status for character error testing.



<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:STATUS:START?
<b>Returns</b>	COUNTING or NOT_COUNTING for START status for character error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: CHARACTER:TEST:STATUS:START? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:STATUS:START COUNTING, indicating that the character error comparisons are being made.

### TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME? (Query Only)

This command queries the elapsed time (in days, hours, minutes, and seconds) for character error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME 0,0,2,14, indicating that the elapsed time is 0 days, 0 hours, 2 hours, and 14 seconds for character error testing.

### TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:DAY? (Query Only)

This command queries the elapsed time days component for character error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:DAY?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: CHARACTER:TEST:TIME:DAY? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME:DAY 0, indicating that the elapsed time days component for character error testing is 0.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:HOURS? (Query Only)

This command queries the elapsed time hours component for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:HOURS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:  
CHARACTER:TEST:TIME:HOURS? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME:HOURS 0,  
indicating that the elapsed time hours component for character error testing is 0.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:MINUTES? (Query Only)

This command queries the elapsed time minutes component for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:MINUTES?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:  
CHARACTER:TEST:TIME:MINUTES? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME:MINUTES  
0, indicating that the elapsed time minutes component for character error  
testing is 0.

## TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:SECONDS? (Query Only)

This command queries the elapsed time seconds component for character error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:CHARACTER:TEST:TIME:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:  
CHARACTER:TEST:TIME:SECONDS?? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:CHARACTER:TEST:TIME:SECONDS  
25, indicating that the elapsed time seconds component for character error  
testing is 25.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:COUNT

This command sets or queries the test duration count as number of bits, frames, symbols, or characters to be tested for error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:COUNT <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:COUNT?

**Arguments** <NR1> indicates the number of bits, frame, symbols, or characters to be tested for the test duration count.

Supports 48bit values. The bit ranges from 0 to 00x7FFFFFFFFFFFF. A value of -1, which is the default value, indicates count forever.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:COUNT 100000000000  
indicates that the test duration count is 100000000000.

TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:COUNT might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:COUNT 100000000000,  
indicates that the test duration count for error testing is set to 100000000000.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:SECONDS

This command sets or queries the test duration in seconds.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:SECONDS <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:SECONDS?

**Arguments** <NR1> is the test duration in seconds.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:SECONDS 30 sets the test duration to 30 seconds.

TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:SECONDS might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:SECONDS 30, indicating that the test duration is 30 seconds.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME

This command sets or queries the test duration time in days, hours, minutes, and seconds.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME?

**Arguments** <NR1> is the test duration time in days, hours, minutes, and seconds. It is in the format <DD, HH, MM, SS>.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME 02,12,59,30 sets the test duration time to two days, 12 hours, 59 minutes, and 30 seconds.

TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME 02,12,59,30, indicating that the test duration time is two days, 12 hours, 59 minutes, and 30 seconds.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:DAYS

This command sets or queries the test duration time days component.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:DAYS <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:DAYS?

**Arguments** <NR1> is a number for the test duration time days component.

- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:DAY 1 sets the test duration time days component for error testing to 1.
- TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:DAY? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:DAY 02, indicating that the test duration time days component is set to 2.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:HOURS

This command sets or queries the test duration time hours component.

- Group** Error Detector
- Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:HOURS <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:HOURS?
- Arguments** <NR1> is a number for the test duration time hours component.
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:HOURS 5 sets the test duration time for error testing to five hours.
- TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:HOURS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:HOURS 05, indicating that the test duration time hours component is set to 5.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:MINUTES

This command sets or queries the test duration time minutes component for error testing.

- Group** Error Detector
- Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:MINUTES <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:MINUTES?
- Arguments** <NR1> is a number for the test duration time minutes component.
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:MINUTES 5 sets the test duration time minutes component for error testing to 5.

TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:MINUTES? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:MINUTES 05, indicating that the test duration time minutes component is set to 5.

## TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:SECONDS

This command sets or queries the test duration time seconds component for error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:SECONDS <NR1>  
TRIGger:A:SERIAL:ERRORdetector:DURATION:TIME:SECONDS?

**Arguments** <NR1> is a number for the test duration time seconds component.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:SECONDS 25 sets the test duration time component for error testing to 25.

TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:DURATION:TIME:SECONDS 25, indicating that the test duration time seconds component is set to 25.

## TRIGger:A:SERIAL:ERRORdetector:ENAbLe

This command sets or queries the status of the error option. ENAbLe must be ON to activate the error detector.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:ENAbLe {ON|OFF}  
TRIGger:A:SERIAL:ERRORdetector:ENAbLe?

**Arguments** ON enables the software error detector feature.

OFF disables the software error detector feature. This is the default.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:ENABLEON enables the error detector option.

TRIGGER:A:SERIAL:ERRORETECTOR:ENABLE might return  
:TRIGGER:A:SERIAL:ERRORETECTOR:TYPE OFF, indicating that the error  
detector option is disabled.

## TRIGger:A:SERIAL:ERRORdetector:ERRORLIMIT

This command sets or queries the error limit value to use when STOPWHEN is  
ERROR.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:ERRORLIMIT <NR1>  
TRIGger:A:SERIAL:ERRORdetector:ERRORLIMIT?

**Arguments** <NR1> is the maximum number of errors.

**Examples** TRIGGER:A:SERIAL:ERRORETECTOR:ERRORLIMIT 2 sets the error limit  
value to 2.

TRIGGER:A:SERIAL:ERRORETECTOR:ERRORLIMIT? might return  
:TRIGGER:A:SERIAL:ERRORETECTOR:ERRORLIMIT 1, indicating that the  
error limit value is 1.

## TRIGger:A:SERIAL:ERRORdetector:FILE:RECALL (No Query Form)

This command initiates a file recall of error detector parameters from a text file.  
Argument is the file name.

The setup files are supplied with the instrument. You can also create your own  
setup files. The setup for the USB CP0\_SKP signal test pattern is supplied in  
“C:\TekScope\ErrorDetector\UsbCP0\_SKPsymbolErrorSetup.txt”.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FILE:RECALL <fileName>

**Arguments** <fileName> is the file name and file location. The default location is  
“C:\TekScope\ErrorDetector” and the setup file is in TXT format.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FILE:RECALL  
 "C:\TekScope\ErrorDetector\SataFrameComplianceFrameErrorSetup.txt"  
 recalls the error detector parameters from the setup file  
 "SataFrameComplianceFrameErrorSetup.txt".

## TRIGger:A:SERIAL:ERRORdetector:FILE:SAVE (No Query Form)

This command initiates a file save of error detector parameters to a text file.  
 Argument is the file name.

The setup files are supplied with the instrument. You can also create your own setup files. The setup for the USB CP0\_SKP signal test pattern is supplied in "C:\TekScope\ErrorDetector\UsbCP0\_SKPsymbolErrorSetup.txt".

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FILE:SAVE <fileName>

**Arguments** <fileName> is the file name and file location. The default location is "C:\TekScope\ErrorDetector" and the setup file is in TXT format.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FILE:SAVE  
 "C:\TekScope\ErrorDetector\SataFrameComplianceFrameErrorSetup.txt"  
 saves the error detector parameters to the setup file  
 "SataFrameComplianceFrameErrorSetup.txt".

## TRIGger:A:SERIAL:ERRORdetector:FRAME? (Query Only)

This command queries all frame error settings, status, and results.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME? might  
 return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:SOFT  
 "00111110101011011110110011101100";EOF  
 "00111110101011011010101110101011";INITIALCRCVALUE  
 1379029042;TEST:STATUS:START STOPPED;LOCK NOT\_LOCKED;SIGNAL  
 NO\_SIGNAL;MAX\_AP MAX\_AP\_OK, indicating the start of frame



is 00111110101011011110110011101100, the end of frame is 00111110101011011010101110101011, the initial CRC value is 1379029042, the test is stopped, not locked, there is no signal, MAX\_AP is ok.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:EOF

This command sets or queries the End of Frame for frame error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:EOF <Qstring> TRIGger:A:SERIAL:ERRORdetector:FRAME:EOF?
<b>Arguments</b>	<Qstring> is a quoted string representing a 32-bit pattern.
<b>Examples</b>	<p>TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:EOF "00111110101011011010101110101011" sets the end of frame to "00111110101011011010101110101011".</p> <p>TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:EOF? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:EOF "00111110101011011010101110101011", indicating that the end of frame is "00111110101011011010101110101011".</p>

## TRIGger:A:SERIAL:ERRORdetector:FRAME:INITIALCRCVALue

This command sets or queries the initial CRC value for frame error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:INITIALCRCVALue <NR1> TRIGger:A:SERIAL:ERRORdetector:FRAME:INITIALCRCVALue?
<b>Arguments</b>	<NR1> is a value defined by the selected standard.
<b>Examples</b>	<p>TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:INITIALCRCVALUE 1379029042 sets the CRC value to 1379029042.</p> <p>TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:INITIALCRCVALUE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:INITIALCRCVALUE</p>

1379029042, indicating that the initial CRC value is set to 1379029042, the SATA default.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:SOF

This command sets or queries the Start of Frame for frame error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:SOF <Qstring> TRIGger:A:SERIAL:ERRORdetector:FRAME:SOF?
<b>Arguments</b>	<Qstring> is a quoted string representing a 32-bit pattern.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:SOF "00111110101011011110110011101100" sets the start of frame.  TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:SOF? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:SOF " 00111110101011011110110011101100", indicating that the start of frame is set to "00111110101011011010101110101011".

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST

This command initiates and terminates frame error testing for the arguments START and STOP. Also zeroes the frame error results for the argument CLEAR.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST <START STOP CLEAR> TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST?
<b>Arguments</b>	START initiates the frame error test counting of errors and duration.  STOP terminates the frame error test counting of frame errors and duration.  CLEAR zeroes the frame error test count, duration, and rate.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST START initiates the frame error test counting.

TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST STOPPED;LOCKED;SIGNAL;MAX\_AP\_OK;0;0;0;14, indicating that the signal is stopped and locked for frame error testing. The status of MAX\_AP is ok and the elapsed duration is 14 seconds.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:BADCHARS? (Query Only)

This command queries the illegal character count for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:BADCHARS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:BADCHARS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:BADCHARS 0, indicating that the illegal character count for the frame error test is 0.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:COUNT? (Query Only)

This command queries the test error count for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:COUNT?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:COUNT? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:COUNT 0, indicating that the test error count for the frame error testing is 0.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DISPARITY? (Query Only)

This command queries the disparity error count for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DISPARITY?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:DISPARITY? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:DISPARITY 0, indicating that the disparity error count for the frame error test is 0.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DURATION? (Query Only)

This command queries the elapsed duration in number of frames.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:DURATION?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:DURATION? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:DURATION 4253659, indicating that the elapsed duration is 4253659 frames.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:MAXALIGNs? (Query Only)

This command queries the result of the maximum consecutive aligns encountered for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:MAXALIGNs?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:MAXALIGNs? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:MAXALIGNs 4, indicating that the maximum consecutive aligns encountered for frame error test is 4.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RATE? (Query Only)

This command queries the calculated frame error rate.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:RATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:RATE 0.0000, indicating that the calculated frame error rate is 0.0000.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RESULTS? (Query Only)

This command queries all the results for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:RESULTS?

**Returns** COUNT is the frame error count (number of bad frames).  
 DURATION is the elapsed duration in units of frames tested.  
 RATE is the calculated frame error rate.  
 MAXALIGNs is the maximum consecutive aligns encountered.  
 BADCHARS is the illegal character count.  
 DISPARITY is the disparity error count.  
 SECONDS is the elapsed duration time in seconds.  
 TIME is the elapsed duration time in DD:HH:MM:SS.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:RESULTS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:RESULTS 0,4253659,0.0000,4,0,0,171,0,0,2,51, indicating that the error count is four, error duration is 4253659, error rate is 0.0000, error maximum consecutive aligns is 4, illegal character count is 0, disparity error count is 0, elapsed duration time is 171 seconds, and elapsed duration time is 0 days, 0 hours, 2 minutes, and 51 seconds.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:SECONDS? (Query Only)

This command queries the result of elapsed duration in seconds for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:SECONDS 20, indicating that the elapsed duration is 20 seconds for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS? (Query Only)

This command queries all of the status for frame error status.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS?

**Returns** COUNTING or NOT\_COUNTING for START status for frame error test.  
LOCKED or NOT\_LOCKED for LOCK status for frame error test.  
SIGNAL or NO\_SIGNAL for SIGNAL status for frame error test.  
MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for frame error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS STOPPED;LOCKED;NO\_SIGNAL;MAX\_AP\_OK, indicating that the frame error status is STOPPED, LOCKED, NO\_SIGNAL, and MAX\_AP\_OK.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:MAX\_AP? (Query Only)

This command queries the MAX\_AP status for the frame error test.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:MAX\_AP?

**Returns** MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for frame error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:MAX\_AP? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:MAX\_AP MAX\_AP\_OK, indicating that the MAX\_AP status is ok for the frame error test.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:LOCK? (Query Only)

This command queries the LOCK status for the frame error test.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:LOCK?
<b>Returns</b>	LOCKED or NOT_LOCKED for LOCK status for frame error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:LOCK? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:LOCK LOCKED, indicating that the frame is LOCKED for the error test.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:SIGNAL? (Query Only)

This command queries the SIGNAL status for the frame error test.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:SIGNAL?
<b>Returns</b>	SIGNAL or NO_SIGNAL for SIGNAL status for frame error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:SIGNAL? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:SIGNAL SIGNAL, indicating that signal is connected for the frame error test.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:START? (Query Only)

This command returns the START status for frame error tests.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:STATUS:START?

**Returns** COUNTING or NOT\_COUNTING for START status for frame error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:START? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:STATUS:START COUNTING, indicating that the START status is COUNTING.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME? (Query Only)

This command queries the elapsed time (in days, hours, minutes, and seconds) for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME 0,0,0,14, indicating that the elapsed time is 0 days, 0 hours, 0 minutes, and 14 seconds for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:DAYs? (Query Only)

This command queries the elapsed time days component for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:DAYs?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:DAYs? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:DAYs 0, indicating that the elapsed time days component is 0 for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:HOuRS? (Query Only)

This command queries the elapsed time hours component for frame error testing.

**Group** Error Detector



**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:HOURS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:HOURS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:HOURS 0, indicating that the elapsed time hours component is 0 for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:MINUTES? (Query Only)

This command queries the elapsed time minutes component for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:MINUTES?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:MINUTES? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:MINUTES 0, indicating that the elapsed time minutes component is 0 for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:SECONDS? (Query Only)

This command queries the elapsed time seconds component for frame error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:TIME:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:SECONDS?? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:FRAME:TEST:TIME:SECONDS 25, indicating that the elapsed time seconds component is 25 for frame error testing.

## TRIGger:A:SERIAL:ERRORdetector:MAXALIGNS

This command sets or queries the maximum consecutive align primitives before a MAX\_AP\_FAIL error is reported.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:MAXALIGNs <NR1>  
TRIGger:A:SERIAL:ERRORdetector:MAXALIGNs?

**Arguments** <NR1> is a integer. The limit values are 0 to 63 and the default is 8.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:MAXALIGNs 6 sets the maximum consecutive align primitives to 6 before an error occurs. Exceeding the maximum limit sets the MAX\_AP\_FAIL status error.

TRIGGER:A:SERIAL:ERRORDETECTOR:MAXALIGNs? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:MAXALIGNs 11, indicating that the maximum consecutive align primitives is 11 before an error occurs. Exceeding the maximum limit sets the MAX\_AP\_FAIL status error.

## TRIGger:A:SERIAL:ERRORdetector:PATTERNNAME

This command sets or queries the pattern name stored in the setup file.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:PATTERNNAME <Qstring>  
TRIGger:A:SERIAL:ERRORdetector:PATTERNNAME?

**Arguments** <Qstring> is a quoted string representing a pattern name.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:PATTERNNAME "FrameComp" sets the pattern name to FrameComp.

TRIGGER:A:SERIAL:ERRORDETECTOR:PATTERNNAME? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:PATTERNNAME "FrameComp",  
indicating that the pattern name is FrameComp. When a setup file is saved or recalled, the pattern name is displayed on the user interface or program interface.

## TRIGger:A:SERIAL:ERRORdetector:SCRAMBLED

This command sets or queries the status of the error detection data scrambling option.

**Group** Error Detector

<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:SCRAMBLED {ON OFF} TRIGger:A:SERIAL:ERRORdetector:SCRAMBLED?
<b>Arguments</b>	ON enables the error detection data scrambling option. This is the default option. OFF disables the error detection data scrambling option.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:SCRAMBLED ON specifies that the error detection data scrambling option is enabled.  TRIGGER:A:SERIAL:ERRORDETECTOR:SCRAMBLED? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SCRAMBLED ON, indicating that the error detection data scrambling option is enabled.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS? (Query Only)

This command queries the RD- skip order set enable and string values.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS?
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS? :TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:PATTERN ON, "00111110100011111010001111101010101010", indicating that the RD- skip order set is enables and the pattern is 00111110100011111010001111101010101010.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:ENABLE

This command sets or queries the status of the RD- skip order set enable option.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS: ENABLE <ON OFF> TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS: ENABLE?

**Arguments** ON enables the RD- skip order set option.  
OFF disables the RD- skip order set option.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:ENABLE ON enables the RD- skip order set option.  
  
TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:ENABLE? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:ENABLE ON, indicating that the RD- skip order set option is enabled.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:PATtern

This command sets or queries the RD- skip order set pattern.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:PATtern <Qstring>  
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:PATtern?

**Arguments** <Qstring> is a quoted string representing a 40-bit pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:PATTERN"00111110100011111010001111101010101010" sets the RD- skip order set pattern.  
  
TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:PATTERN? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDMINUS:PATTERN "00111110100011111010001111101010101010", indicating that the RD- skip order set pattern is set.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS? (Query Only)

This command queries the RD+ skip order set enable and string values.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS?  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:PATTERN  
ON, “00111110100011111010001111101010101010”,  
indicating that the RD+ skip order set is enabled and the pattern is  
00111110100011111010001111101010101010.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:ENABLE

This command sets or queries the status of the RD+ skip order set enable option.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:  
ENABLE <ON|OFF>  
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:  
ENABLE?

**Arguments** ON enables the RD+ skip order set option.  
OFF disables the RD+ skip order set option.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:  
ENABLE ON, enables the RD+ skip order set enable option.  
  
TRIGGER:A:SERIAL:ERRORDETECTOR:  
SKIPSETPRIMITIVE:RDPLUS:ENABLE? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:ENABLE  
ON, indicating that the RD+ skip order set is enabled.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:PATtern

This command sets or queries the RD+ skip order set pattern.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:  
PATtern <Qstring>  
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:  
PATtern?

**Arguments** <Qstring> is a quoted string representing a 40-bit pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:PATTERN "00111110100011111010001111101010101010", sets the RD+ skip order set pattern.

TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:PATTERN? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:RDPLUS:PATTERN "00111110100011111010001111101010101010", indicating that the RD+ skip order set pattern is set.

## TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:S8BPATtern

This command sets or queries the 32-bit skip order set pattern.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:S8BPATtern <string>  
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:S8BPATtern?

**Arguments** <Qstring> is a quoted string representing a 32-bit pattern.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:S8BPATTERN "00111101010100100101001011011110" sets the 32-bit skip order set pattern to "00111101010100100101001011011110".

TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:S8BPATTERN? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SKIPSETPRIMITIVE:S8BPATTERN "00111101010100100101001011011110", indicating that the 32-bit skip order set pattern is "00111101010100100101001011011110".

## TRIGger:A:SERIAL:ERRORdetector:SSC

This command sets or queries the status of the spread spectrum clock tracking option.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SSC {ON|OFF}  
TRIGger:A:SERIAL:ERRORdetector:SSC?

- Arguments** ON enables spread spectrum clock tracking. For error detector, the spread spectrum clock tracking should always be turned on.
- OFF disables spread spectrum clock tracking. For serial trigger, the spread spectrum clock tracking is turned off.
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SSC ON enables spread spectrum clock tracking option.
- TRIGGER:A:SERIAL:ERRORDETECTOR:SSC? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SSC ON, indicating that spread spectrum clock tracking is enabled.

## TRIGger:A:SERIAL:ERRORdetector:STANDARD

This command sets or queries the standard selection for error testing.

- Group** Error Detector
- Syntax** TRIGger:A:SERIAL:ERRORdetector:STANDARD <LIST>  
TRIGger:A:SERIAL:ERRORdetector:STANDARD?
- Arguments** <LIST> is the supported standard. The supported standards are SATA, USB, PCIE, and ANY8B10B.
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:STANDARD SATA sets the standard to SATA.
- TRIGGER:A:SERIAL:ERRORDETECTOR:STANDARD? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:STANDARD SATA, indicating that the selected standard is SATA.

## TRIGger:A:SERIAL:ERRORdetector:STOPWHEN

This command sets or queries the stopping condition. The test can be stopped when the count, time, or number of errors elapses. If the STOPWHEN value is MANUAL, the test runs until a TEST STOP command is received.

- Group** Error Detector
- Syntax** TRIGger:A:SERIAL:ERRORdetector:STOPWHEN  
<MANUAL | COUNT | TIME | ERROR>

TRIGger:A:SERIAL:ERRORdetector:STOPWHEN?

- Arguments** MANUAL indicates that the test must be stopped by issuing a TEST STOP command. This is the default.
- COUNT stops the test when DURATION:COUNT comparisons has been made. The comparisons may be bit, frame, symbol, or character as appropriate for the TEST:TYPE.
- TIME stops the test when DURATION:TIME elapses.
- ERROR stops the test when the number of errors  $\geq$  ERRORLIMIT.
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:STOPWHEN MANUAL requires the stop test action to stop the test.
- TRIGGER:A:SERIAL:ERRORDETECTOR:STOPWHEN? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:STOPWHEN MANUAL, indicating that the stop test action must be used to stop the test.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL? (Query Only)

This command queries all symbol error settings, status, and results.

- Group** Error Detector
- Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL?
- Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:START STOPPED;LOCK NOT\_LOCKED;SIGNAL NO\_SIGNAL;MAX\_AP MAX\_AP\_OK;:TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:DAY 0;HOURS 0;MINUTES 3;SECONDS 0, that the test is stopped, not locked, there is no signal, MAX\_AP is ok, and the test time is 3 minutes.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST

This command initiates and terminates symbol error testing for the arguments START and STOP. Also zeroes the symbol error results for the argument CLEAR. This command also queries all of the symbol test settings and results for symbol error testing.

- Group** Error Detector



<b>Syntax</b>	<pre>TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST &lt;START STOP CLEAR&gt; TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST?</pre>
<b>Arguments</b>	<p>START initiates symbol and bit error testing.</p> <p>STOP terminates symbol and bit error testing.</p> <p>CLEAR zeroes the symbol and bit error counts, duration, bit error rate, and symbol error rate.</p>
<b>Examples</b>	<p>TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST START initiates symbol and bit error testing.</p> <p>TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST STOPPED;LOCKED;SIGNAL;MAX_AP_OK;0,0,0,14, indicating that the signal is stopped and locked for symbol and bit error testing. The status of MAX_AP is ok and the elapsed duration is 0 days, 0 hours, 0 minutes, and 14 seconds.</p>

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BADCHARS? (Query Only)

This command queries the illegal character count for symbol error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	<pre>TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BADCHARS?</pre>
<b>Examples</b>	<p>TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BADCHARS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BADCHARS 0, indicating that the illegal character count for the symbol error test is 0.</p>

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITCOUNT? (Query Only)

This command queries the bit error count (number of bad bits) for symbol error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	<pre>TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITCOUNT?</pre>

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITCOUNT? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITCOUNT 0, indicating that the bit error count for the symbol error test is 0.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITDURATION? (Query Only)

This command queries the elapsed duration in units of bits tested for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITDURATION?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITDURATION? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITDURATION 24, indicating that the elapsed duration tested for the symbol error test is 24 bits.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITRATE? (Query Only)

This command queries the calculated bit error rate for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:BITRATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITRATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:BITRATE 0.0000, indicating that the calculated bit error rate for symbol error testing is 0.0000.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:COUNT? (Query Only)

This command queries the symbol error count for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:COUNT?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:COUNT? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:COUNT 0, indicating that the symbol error count is 0.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DISPARITY? (Query Only)

This command queries the disparity error count for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DISPARITY?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:DISPARITY? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:DISPARITY 0, indicating that the disparity error count for the symbol error test is 0.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DURATION? (Query Only)

This command queries the elapsed duration (in units of symbols) for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:DURATION?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:DURATION? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:DURATION 12634, indicating that the elapsed duration in units of symbols for the symbol error test is 12634.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:MAXALIGN? (Query Only)

This command queries the maximum consecutive skip order sets encountered for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:MAXALIGN?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:MAXALIGN? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:MAXALIGN 3, indicating that the maximum consecutive skip order sets encountered for the symbol error test is 3.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RATE? (Query Only)

This command queries the calculated symbol error rate for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RATE?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:RATE? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:RATE 0.0000, indicating that the calculated symbol error rate tested is 0.0000.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RESULTS? (Query Only)

This command queries all of the results for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:RESULTS?

**Returns**

- COUNT which is the symbol error count (number of bad 10-bit symbols).
- DURATION which is the elapsed duration in units of symbols tested.
- RATE which is the calculated symbol error rate.
- MAXALIGNS which is the maximum consecutive skipsets encountered.
- BADCHARS which is the illegal character count.
- DISPARITY which is the disparity error count.
- BITCOUNT is the USB bit error count (number of bad bits).
- BITDURATION is the elapsed duration in units of bits tested.
- BITRATE is the calculated USB bit error rate.
- SECONDS is the elapsed duration time in seconds.

TIME is the elapsed duration time in DD:HH:MM:SS.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:RESULTS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:RESULTS 0, 12657, 0.0000, 3, 0, 0, 0, 0.0000, 0,0,1,20, indicating that the error count is 0, elapsed duration is 12657, error rate is 0.0000, maximum consecutive skipsets encountered is 3, illegal character count is 0, disparity error count is 0, USB bit error count is 0, elapsed duration is 36, calculated USB bit error rate is 0.0000, elapsed duration time is 10 seconds, and elapsed duration time is 0 days, 0 hours, 10 minutes, and 20 seconds.

### TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:SECONDS? (Query Only)

This command queries the elapsed duration time (in seconds) for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:SECONDS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:SECONDS 45, indicating that the elapsed duration time for symbol error testing is 45 seconds.

### TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS? (Query Only)

This command queries all of the status for the symbol error test.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS?

**Returns** COUNTING or NOT\_COUNTING for START status for character error test.  
 LOCKED or NOT\_LOCKED for LOCK status for character error test.  
 SIGNAL or NO\_SIGNAL for SIGNAL status for character error test.  
 MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for character error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS STOPPED;LOCKED;NO\_SIGNAL;MAX\_AP\_OK, indicating that all of the status for the symbol error test are STOPPED, LOCKED, NO\_SIGNAL, and MAX\_AP\_OK.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:MAX\_AP? (Query Only)

This command queries the MAX\_AP status for the symbol error test.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:MAX\_AP?

**Returns** MAX\_AP\_OK or MAX\_AP\_FAIL for MAX\_AP status for character error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:MAX\_AP? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:MAX\_AP MAX\_AP\_OK, indicating that the MAX\_AP status is ok for the symbol error test.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:LOCK? (Query Only)

This command queries the LOCK status for the symbol error test.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:LOCK?

**Returns** LOCKED or NOT\_LOCKED for LOCK status for character error test.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:LOCK? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:LOCK LOCKED, indicating that the recovered reference clock is LOCKED for the symbol error test.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:SIGNAL? (Query Only)

This command queries the SIGNAL status for the symbol error test.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:SIGNAL?
<b>Returns</b>	SIGNAL or NO_SIGNAL for SIGNAL status for character error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: SYMBOL:TEST:STATUS:SIGNAL? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:SIGNAL SIGNAL, indicating that the signal is connected for the symbol error test.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:START? (Query Only)

This command queries the START status for symbol error testing.

<b>Group</b>	Error Detector
<b>Syntax</b>	TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:STATUS:START?
<b>Returns</b>	COUNTING or NOT_COUNTING for START status for character error test.
<b>Examples</b>	TRIGGER:A:SERIAL:ERRORDETECTOR: SYMBOL:TEST:STATUS:START? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:STATUS:START COUNTING, indicating that the symbol error test is comparing symbols and counting any errors that occur.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME? (Query Only)

This command queries the elapsed time (in days, hours, minutes, and seconds) for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME 0,0,0,14, indicating that the elapsed time is 0 days, 0 hours, 0 minutes, and 14 seconds for symbol error testing.

### TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:DAYs? (Query Only)

This command queries the elapsed time days component for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:DAYs?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:DAYs? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:DAYs 0, indicating that the elapsed time days component is 0 for symbol error testing.

### TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:HOuRS? (Query Only)

This command queries the elapsed time hours component for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:HOuRS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:HOuRS? might return :TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:HOuRS 0, indicating that the elapsed time hours component is 0 for symbol error testing.

### TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:MINuTES? (Query Only)

This command queries the elapsed time minutes component for symbol error testing.



**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:MINUTES?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:  
SYMBOL:TEST:TIME:MINUTES? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:MINUTES 0,  
indicating that the elapsed time minutes component is 0 for symbol error testing.

## TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:SECONDS? (Query Only)

This command queries the elapsed time seconds component for symbol error testing.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:SYMBOL:TEST:TIME:SECONDS?

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:  
SYMBOL:TEST:TIME:SECONDS?? might return  
:TRIGGER:A:SERIAL:ERRORDETECTOR:SYMBOL:TEST:TIME:SECONDS 25,  
indicating that the elapsed time seconds component is 25 for symbol error testing.

## TRIGger:A:SERIAL:ERRORdetector:TYPE

This command sets or queries the error detector type.

**Group** Error Detector

**Syntax** TRIGger:A:SERIAL:ERRORdetector:TYPE  
{BIT|FRAME|SYMBOL|CHARACTER}  
TRIGger:A:SERIAL:ERRORdetector:TYPE?

**Arguments** BIT sets the error detector type to bit.  
FRAME sets the error detector type to frame.  
SYMBOL sets the error detector type to symbol.

CHARACTER sets the error detector type to character.

**Examples** TRIGGER:A:SERIAL:ERRORDETECTOR:TYPE BIT sets the error detector type to bit.

TRIGGER:A:SERIAL:ERRORDETECTOR:TYPE might return :TRIGGER:A:SERIAL:ERRORDETECTOR:TYPE BIT, indicating that the error detector type is bit.

## TRIGger:A:SERIAL:LOCKLen

This command sets or queries the length in bits of the repeating bit pattern for pattern lock trigger.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:LOCKLen <NR1>  
TRIGger:A:SERIAL:LOCKLen?

**Arguments** <NR1> specifies the length in bits. The valid values are 1 to 2147483647.

**Examples** TRIGGER:A:SERIAL:LOCKLEN2640 sets the repeating bit pattern length to 2640 bits.

TRIGGER:A:SERIAL:LOCKLEN? might return :TRIGGER:A:SERIAL:LOCKLEN 10000, indicating that the pattern string is 10000 bits.

## TRIGger:A:SERIAL:LOCKOffset

This command sets or queries the current bit offset into the pattern lock trigger bit pattern.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:LOCKOffset <NR1>  
TRIGger:A:SERIAL:LOCKOffset?

**Arguments** <NR1> specifies the bit offset. The valid values are 1 to 2147483646.

**Examples** TRIGGER:A:SERIAL:LOCKOFFSET10 sets the current bit offset to 10 bits.  
 TRIGGER:A:SERIAL:LOCKOFFSET? might return  
 :TRIGGER:A:SERIAL:LOCKOFFSET 20, indicating that the current bit offset  
 is 20 bits.

## TRIGger:A:SERIAL:SOUrce

This command sets or queries the serial data source channel.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:SOUrce {CH1|CH2|CH3|CH4}  
 TRIGger:A:SERIAL:SOUrce?

**Arguments** CH1-CH4 specifies one of the input channels.

**Examples** TRIGGER:A:SERIAL:SOURCE CH1 sets the serial data source channel to CH1.  
 TRIGGER:A:SERIAL:SOURCE? might return :TRIGGER:A:SERIAL:SOURCE  
 CH4, indicating that the serial data source channel is CH4.

## TRIGger:A:SERIAL:STANdard

This command sets or queries the standard that identifies the code and bit rate.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:STANdard  
 {FC133|FC266|FC531|FC1063|D1|D2|VIDEO270|VIDEO360|OC1|OC3|  
 OC12|ENET1250|FW1394BS400B|FW1394BS800B|CUSTOM|ENET100FX|  
 RIO\_500M|RIO\_750M|RIO\_1G|RIO\_SERIAL\_1G|VSROC192|ENETXAUI|  
 SAS3\_?|PCIExpress|INFINIBAND|RIO\_SERIAL\_2G|RIO\_SERIAL\_3G|  
 FC2125|RIO\_2G|FW1394BS1600B|SAS1\_5|ENETXAUI|ENETXAUI2|  
 FC2125|FC4250|FW1394BS1600B|INFINIBAND|PCIExpress|PCIExpress2|  
 RIO\_2G|RIO\_500M|RIO\_750M|RIO\_SERIAL\_1G|RIO\_SERIAL\_2\_5G|  
 RIO\_SERIAL\_3G|SAS6\_0|SATA1\_5|SATA3\_0|SATA6\_0}  
 TRIGger:A:SERIAL:STANdard?

## Related Commands

Arguments	
	FC133: NRZ, 132.8 Mb/s
	FC266: NRZ, 265.6 Mb/s
	FC531: NRZ, 531.2 Mb/s
	FC1063: NRZ, 1.063 Gb/s
	OC1: NRZ, 51.84 Mb/s
	OC3: NRZ, 155.5 Mb/s
	OC12: NRZ, 622.1 Mb/s
	ENET1250: NRZ, 3.125 Gb/s
	ENETXAUI: NRZ, 1.25 Gb/s
	ENETXAUI2: NRZ, 6.25 Gb/s
	FW1394BS400B: NRZ, 491.5 Mb/s
	FW1394BS1600B: NRZ, 1.966 Gb/s
	FW1394BS800B: NRZ, 983.0 Mb/s
	CUSTOM: NRZ, 155.5 Mb/s
	ENET100FX: MLT3, 100 Mb/s
	RIO_500M: NRZ, 500 Mb/s, triggers RIO_DRV500, RIO_EDRV500M, and RIO_RCV500, 500Mb/s
	RIO_750M: NRZ, 750 Mb/s, triggers RIO_DRV750, RIO_EDRV750M, and RIO_RCV750, 750Mb/s
	RIO_1G: NRZ, 1.0 Gb/s, triggers RIO_1G
	RIO_SERIAL_1G: NRZ, 1.25 Gb/s, triggers RIO_Serial_1G
	RIO_2G: NRZ, 2.0Gb/s
	RIO_SERIAL_2_5G: NRZ, 2.5 Gb/s, triggers RIO_Serial_1G
	RIO_SERIAL_3G: NRZ, 3.0 Gb/s, triggers RIO_Serial_1G
	VSROC192: NRZ, 1.2441 Gb/s, triggers VSR OC192/STM64
	All of the standards may not be available on every instrument.
	FC2125: NRZ, 2.125 Gb/s
	FC4250: NRZ, 4.25 Gb/s
	INFINIBAND: NRZ, 2.5 Gb/s
	PCIExpress: NRZ, 2.5 Gb/s

PCIExpress2: NRZ, 5.0 Gb/s

SAS6\_0: NRZ, SAS6\_0

SATA1\_5: NRZ, 1.5 Gb/s

SATA3\_0: NRZ, 3.0 Gb/s

SATA6\_0: NRZ, 6.0 Gb/s

**Examples** TRIGGER:A:SERIAL:STANDARD OC1 sets the standard that identifies the code and bit rate to NRZ, 51.84 Mb/s.

TRIGGER:A:SERIAL:STANDARD? might return  
:TRIGGER:A:SERIAL:STANDARD CUSTOM, indicating that the standard that identifies the code and bit rate is set to NRZ, 155.5 Mb/s.

## TRIGger:A:SERIAL:TRIGgeron

This command sets or queries the trigger on a designated arbitrary bit pattern or lock on a repeating pattern of known length.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SERIAL:TRIGgeron {PATtern|LOck}  
TRIGger:A:SERIAL:TRIGgeron?

**Arguments** PATtern specifies the trigger on a designated arbitrary bit pattern.  
LOck specifies the trigger locks a repeating pattern of known length.

**Examples** TRIGGER:A:SERIAL:TRIGGERON PATTERN sets trigger on a designated arbitrary bit pattern.

TRIGGER:A:SERIAL:TRIGGERON? might return  
:TRIGGER:A:SERIAL:TRIGGERON LOCK, indicating that the trigger locks a repeating pattern of known length.

## TRIGger:A:SPI:CONDition

DPO7000 Series only: This command sets or queries the trigger condition for SPI triggering.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:CONDition {MISO|MOSI|MISOMOSI}  
TRIGger:A:SPI:CONDition?

**Arguments** MISO specifies the MISO condition for SPI triggering.  
MOSI specifies the MOSI condition for SPI triggering.  
MISOMOSI specifies either the MISO or MOSI condition for SPI triggering.

**Examples** TRIGGER:A:SPI:CONDITION MOSI sets the SPI trigger condition to MOSI.  
TRIGGER:A:SPI:CONDITION? might return :TRIGGER:A:SPI:CONDITION  
MOSI, indicating that the SPI trigger condition is set to MOSI.

## TRIGger:A:SPI:DATA:MISO:ACTIVE

DPO7000 Series only: This command sets or queries the SPI MISO polarity.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:DATA:MISO:ACTIVE {HIGH|LOW}  
TRIGger:A:SPI:DATA:MISO:ACTIVE?

**Arguments** HIGH specifies HIGH polarity.  
LOW specifies LOW polarity.

**Examples** TRIGGER:A:SPI:DATA:MISO:ACTIVE HIGH sets the SPI trigger MISO polarity to high.  
TRIGGER:A:SPI:DATA:MISO:ACTIVE? might return  
:TRIGGER:A:SPI:DATA:MISO:ACTIVE LOW, indicating that the SPI MISO polarity is set to low.

## TRIGger:A:SPI:DATA:MISO:LEVEL

DPO7000 Series only: This command sets or queries the threshold for the SPI MISO data source.

**Group** Low Speed Serial Trigger

<b>Syntax</b>	TRIGger:A:SPI:DATA:MISO:LEVel <NR3> TRIGger:A:SPI:DATA:MISO:LEVel?
<b>Arguments</b>	<NR3> specifies the threshold for the SPI trigger MISO data source. The range is $\pm 5.0$ .
<b>Examples</b>	TRIGGER:A:SPI:DATA:MISO:LEVEL 0.0 sets the threshold for the SPI trigger MISO data source to 0.0.  TRIGGER:A:SPI:DATA:MISO:LEVEL? might return :TRIGGER:A:SPI:DATA:MISO:LEVEL 5.0, indicating that the threshold for the SPI MISO data source is set to 5.0.

## TRIGger:A:SPI:DATA:MISO:SOURce

DPO7000 Series only: This command sets or queries the MISO data source for the SPI trigger.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SPI:DATA:MISO:SOURce CH<x> TRIGger:A:SPI:DATA:MISO:SOURce?
<b>Arguments</b>	CH<x> specifies the source for the SPI trigger MISO data source. x can be 1, 2, 3, or 4.
<b>Examples</b>	TRIGGER:A:SPI:DATA:MISO:SOURCE CH2 sets the SPI trigger MISO data source to channel CH2.  TRIGGER:A:SPI:DATA:MISO:SOURCE? might return :TRIGGER:A:SPI:DATA:MISO:SOURCE CH1, indicating that the SPI trigger MISO data source is set to CH1.

## TRIGger:A:SPI:DATA:MISO:VALue

DPO7000 Series only: This command sets or queries the binary data string used for the SPI trigger if the trigger condition is set to MISO or MISOMOSI.

<b>Group</b>	Low Speed Serial Trigger
--------------	--------------------------

**Syntax** TRIGger:A:SPI:DATA:MISO:VALue <QString>  
TRIGger:A:SPI:DATA:MISO:VALue?

**Arguments** <QString> specifies the binary data string for the SPI data. You can specify up to 32 bits.

**Examples** TRIGGER:A:SPI:DATA:MISO:VALUE "1011" sets the binary data string used for the SPI trigger to 1011.

TRIGGER:A:SPI:DATA:MISO:VALUE? might return  
:TRIGGER:A:SPI:DATA:MISO:VALUE "10111", indicating that the binary data string used for the SPI trigger is set to 10111.

## TRIGger:A:SPI:DATA:MOSI:ACTIVE

DPO7000 Series only: This command sets or queries the SPI MOSI polarity.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:DATA:MOSI:ACTIVE {HIGH|LOW}  
TRIGger:A:SPI:DATA:MOSI:ACTIVE?

**Arguments** HIGH specifies HIGH polarity.  
LOW specifies LOW polarity.

**Examples** TRIGGER:A:SPI:DATA:MOSI:ACTIVE HIGH sets the SPI trigger MOSI polarity to HIGH.

TRIGGER:A:SPI:DATA:MOSI:ACTIVE? might return  
:TRIGGER:A:SPI:DATA:MOSI:ACTIVE LOW, indicating that the SPI MOSI polarity is set to LOW.

## TRIGger:A:SPI:DATA:MOSI:LEVEL

DPO7000 Series only: This command sets or queries the threshold for the SPI MOSI data source.

**Group** Low Speed Serial Trigger



<b>Syntax</b>	TRIGger:A:SPI:DATA:MOSI:LEVel <NR3> TRIGger:A:SPI:DATA:MOSI:LEVel?
<b>Arguments</b>	<NR3> specifies the threshold for the SPI trigger MOSI data source. The range is $\pm 5.0$ .
<b>Examples</b>	TRIGGER:A:SPI:DATA:MOSI:LEVEL 0.0 sets the threshold for the SPI trigger MOSI data source to 0.0.  TRIGGER:A:SPI:DATA:MISO:LEVEL? might return :TRIGGER:A:SPI:DATA:MISO:LEVEL 5.0, indicating that the threshold for the SPI MOSI data source is set to 5.0.

## TRIGger:A:SPI:DATA:MOSI:SOURce

DPO7000 Series only: This command sets or queries the MOSI data source for the SPI trigger.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SPI:DATA:MOSI:SOURce CH<x> TRIGger:A:SPI:DATA:MOSI:SOURce?
<b>Arguments</b>	CH<x> specifies the source for the SPI trigger MOSI data source. x can be 1, 2, 3, or 4.
<b>Examples</b>	TRIGGER:A:SPI:DATA:MOSI:SOURCE CH2 sets the SPI trigger MOSI data source to channel CH2.  TRIGGER:A:SPI:DATA:MOSI:SOURCE? might return :TRIGGER:A:SPI:DATA:MOSI:SOURCE CH1, indicating that the SPI trigger MOSI data source is set to channel CH1.

## TRIGger:A:SPI:DATA:MOSI:VALue

DPO7000 Series only: This command sets or queries the binary data string used for the SPI trigger if the trigger condition is set to MOSI or MISOMOSI.

<b>Group</b>	Low Speed Serial Trigger
--------------	--------------------------

**Syntax** TRIGger:A:SPI:DATA:MOSI:VALue <QString>  
TRIGger:A:SPI:DATA:MOSI:VALue?

**Arguments** <QString> specifies the binary data string for the SPI data. You can specify up to 32 bits.

**Examples** TRIGGER:A:SPI:DATA:MOSI:VALUE "1011" sets the binary data string used for the SPI trigger to 1011.

TRIGGER:A:SPI:DATA:MOSI:VALUE? might return :TRIGGER:A:SPI:DATA:MOSI:VALUE "10111", indicating that the binary data string used for the SPI trigger is set to 10111.

## TRIGger:A:SPI:DATA:START

DPO7000 Series only: This command sets or queries the nth data byte on the data source after the signal on the enable slave source switches to the polarity specified by the Slave Select Polarity. The instrument triggers if the pattern Matches beginning at this point.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:DATA:START <NR3>  
TRIGger:A:SPI:DATA:START?

**Arguments** <NR3> specifies the SPI start byte data. The data can range from 0 to  $2^{16}-1$ .

**Examples** TRIGGER:A:SPI:DATA:START 0 sets the SPI trigger data start to 0.

TRIGGER:A:SPI:DATA:START? might return :TRIGGER:A:SPI:DATA:START 1, indicating that the SPI trigger data is set to 1.

## TRIGger:A:SPI:FORMat

DPO7000 Series only: This command sets or queries the SPI trigger data format.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:FORMat {BINary|HEX}  
TRIGger:A:SPI:FORMat?

- Arguments**    `BINARY` specifies a binary data format.  
                   `HEX` specifies a hexadecimal data format.
- Examples**     `TRIGGER:A:SPI:FORMAT BINARY` sets the SPI trigger data format to `BINARY`.  
                   `TRIGGER:A:SPI:FORMAT?` might return `:TRIGGER:A:SPI:FORMAT 1`,  
                   indicating that the SPI trigger data format is set to `HEX`.

## TRIGger:A:SPI:SCLK:ACTIVE

DPO7000 Series only: This command sets or queries the SPI SCLK polarity.

- Group**        Low Speed Serial Trigger
- Syntax**       `TRIGger:A:SPI:SCLK:ACTIVE {RISe|FALL}`  
                   `TRIGger:A:SPI:SCLK:ACTIVE?`
- Arguments**    `RISe` specifies rising clock polarity.  
                   `FALL` specifies falling clock polarity.
- Examples**     `TRIGGER:A:SPI:SCLK:ACTIVE FALL` sets the SPI SCLK polarity to fall.  
                   `TRIGGER:A:SPI:SCLK:ACTIVE?` might return  
                   `:TRIGGER:A:SPI:SCLK:ACTIVE FALL`, indicating that the SPI SCLK polarity  
                   is set to `FALL`.

## TRIGger:A:SPI:SCLK:LEVel

DPO7000 Series only: This command sets or queries the threshold for the SPI trigger SCLK.

- Group**        Low Speed Serial Trigger
- Syntax**       `TRIGger:A:SPI:SCLK:LEVel <NR3>`  
                   `TRIGger:A:SPI:SCLK:LEVel?`
- Arguments**    `<NR3>` specifies the SPI trigger SCLK threshold.

**Examples** TRIGGER:A:SPI:SCLK:LEVEL 0.0 sets the SPI SCLK threshold to 0.0.

TRIGGER:A:SPI:SCLK:LEVEL? might return :TRIGGER:A:SPI:SCLK:LEVEL 5.0, indicating that the SPI SCLK threshold is set to 5.0.

## TRIGger:A:SPI:SCLK:SOURce

DPO7000 Series only: This command sets or queries the SPI SCLK source.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:SCLK:SOURce CH<x>  
TRIGger:A:SPI:SCLK:SOURce?

**Arguments** CH<x> specifies the SPI SCLK source. x can be 1, 2, 3, or 4.

**Examples** TRIGGER:A:SPI:SCLK:SOURCE CH2 sets the SPI SCLK trigger source to channel CH2.

TRIGGER:A:SPI:SCLK:SOURCE? might return  
:TRIGGER:A:SPI:SCLK:SOURCE CH4, indicating that the SPI SCLK trigger source is set to channel CH4.

## TRIGger:A:SPI:SS:ACTIVE

DPO7000 Series only: This command sets or queries the SPI trigger Slave Select (SS) polarity.

**Group** Low Speed Serial Trigger

**Syntax** TRIGger:A:SPI:SS:ACTIVE {HIGH|LOW}  
TRIGger:A:SPI:SS:ACTIVE?

**Arguments** HIGH specifies HIGH for SS polarity.  
LOW specifies LOW for SS polarity.

**Examples** TRIGGER:A:SPI:SS:ACTIVE HIGH sets the SPI trigger SS polarity to HIGH.

TRIGGER:A:SPI:SS:ACTIVE? might return :TRIGGER:A:SPI:SS:ACTIVE LOW, indicating that the SPI trigger SS polarity is set to LOW.

## TRIGger:A:SPI:SS:LEVel

DPO7000 Series only: This command sets or queries the threshold for the SPI trigger Slave Select (SS) signal.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SPI:SS:LEVel <NR3> TRIGger:A:SPI:SS:LEVel?
<b>Arguments</b>	<NR3> specifies the threshold for the SPI trigger SS line. The threshold range is $\pm 5.0$ .
<b>Examples</b>	TRIGGER:A:SPI:SS:LEVEL0.0 sets the threshold level for the SPI trigger SS to 0.0.  TRIGGER:A:SPI:SS:LEVEL? might return :TRIGGER:A:SPI:SS:LEVEL 5.0000, indicating that the threshold level for the SPI trigger SS is set to 5.0.

## TRIGger:A:SPI:SS:SOUrce

DPO7000 Series only: This command sets or queries the SPI trigger Slave Select (SS) source.

<b>Group</b>	Low Speed Serial Trigger
<b>Syntax</b>	TRIGger:A:SPI:SS:SOUrce CH<x> TRIGger:A:SPI:SS:SOUrce?
<b>Arguments</b>	CH<x> specifies the source for the SPI SS signal. x can be 1, 2, 3, or 4.
<b>Examples</b>	TRIGGER:A:SPI:SS:SOURCECH2 sets the SPI trigger Slave Select (SS) source to channel CH2.  TRIGGER:A:SPI:SS:SOURCE? might return :TRIGGER:A:SPI:SS:SOURCE CH1, indicating that the SPI trigger Slave Select (SS) source is set to channel CH1.

## TRIGger:A:VIDeo? (Query Only)

DPO7000 Series only: This query-only command returns the A trigger video parameters.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo?
<b>Examples</b>	TRIGGER:A:VIDEO? might return :TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED; SCAN RATE1; :TRIGGER:A:VIDEO:FIELD ALLFIELDS; HOLDOFF:FIELD 1.0000;TIME 20.0000E-3; :TRIGGER:A:VIDEO:LINE 1;SCAN RATE1; SOURCE CH1;STANDARD NTSC.

## TRIGger:A:VIDeo:CUSTom? (Query Only)

DPO7000 Series only: This query-only command returns the A trigger custom video parameters.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom?
<b>Examples</b>	TRIGGER:A:VIDEO:CUSTOM? might return the parameters :TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED; SCAN RATE1.

## TRIGger:A:VIDeo:CUSTom:FORMat

DPO7000 Series only: This command sets or queries the video trigger format. Use this command only when the video format is set to custom.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom:FORMat {INTERLACed PROGRESSive} TRIGger:A:VIDeo:CUSTom:FORMat?
<b>Arguments</b>	INTERLACed argument sets the format for interlaced video lines. PROGRESSive argument sets the format for progressive video lines.
<b>Examples</b>	TRIGGER:A:VIDEO:CUSTOM:FORMATPROGRESSIVE sets the custom format for the A video trigger to progressive lines.

TRIGGER:A:VIDEO:CUSTOM:FORMAT? might return  
:TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED, indicating that interlaced  
is selected as the custom format for the A video trigger.

## TRIGger:A:VIDeo:CUSTom:LINEPeriod

DPO7000 Series only: This command sets or queries the period horizontal line scan. Use this command only when the video format is set to Bilevel or Trilevel. This is equivalent to selecting Video Setup from the Trig menu, setting Bilevel or Trilevel for Format, and setting the Line Period.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom:LINEPeriod <NR3> TRIGger:A:VIDeo:CUSTom:LINEPeriod?
<b>Arguments</b>	<NR3> is the time for each horizontal scan line.
<b>Examples</b>	TRIGGER:A:VIDEO:CUSTOM:LINEPERIOD 63.6E-6 sets the custom line period to 63.6 microseconds.  TRIGGER:A:VIDEO:CUSTOM:LINEPERIOD? might return TRIGGER:A:VIDEO:CUSTOM:LINEPERIOD 63.6000E-6 indication the line period is set to 63 microseconds.

## TRIGger:A:VIDeo:CUSTom:SYNCInterval

DPO7000 Series only: This command sets or queries the horizontal line scan rate of the A video trigger. Use this command only when the video format is set to Bilevel. This is equivalent to selecting Video Setup from the Trig menu, setting Bilevel for Format, and setting the sync interval.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom:SYNCInterval <NR3> TRIGger:A:VIDeo:CUSTom:SYNCInterval?
<b>Arguments</b>	<NR3> is the sync interval for the Bilevel format.

- Examples** TRIGGER:A:VIDEO:CUSTOM:SYNCINTERVAL 4.7E-6 sets the sync interval to 4.7 microseconds.
- TRIGGER:A:VIDEO:CUSTOM:SYNCINTERVAL? might return TRIGGER:A:VIDEO:CUSTOM:SYNCINTERVAL 4.7000E-6, indicating that the custom sync interval is set to 4.7 microseconds.

## TRIGger:A:VIDeo:FIELD

DPO7000 Series only: This command sets or queries the video field or line that the trigger detects.

**Group** Trigger

**Syntax** TRIGger:A:VIDeo:FIELD  
{ODD|EVEN|FIELD1|FIELD2|ALLFiElds|ALLLines|NUMERic}  
TRIGger:A:VIDeo:FIELD?

- Arguments** ODD argument sets the instrument to trigger on interlaced video odd fields.
- EVEN argument sets the instrument to trigger on interlaced video even fields.
- FIELD1 argument sets the instrument to trigger on interlaced video odd fields (same as ODD).
- FIELD2 argument sets the instrument to trigger on interlaced video even fields (same as EVEN).
- ALLFiElds argument sets the instrument to trigger on all fields.
- ALLLines argument sets the instrument to trigger on all video lines.
- NUMERic argument sets the instrument to trigger on the video signal line specified by the TRIGger:A:VIDeo:LINE command.

- Examples** TRIGGER:A:VIDEO:FIELD EVEN sets the A video trigger so that it will trigger on even fields.
- TRIGGER:A:VIDEO:FIELD? might return :TRIGGER:A:VIDEO:FIELD ALLFIELDS, indicating that the A video will trigger on all video fields.

## TRIGger:A:VIDeo:HOLdoff:FIELD

DPO7000 Series only: This command sets or queries the video trigger holdoff in terms of video fields.



<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:HOLdoff:FIELD <NR3> TRIGger:A:VIDeo:HOLdoff:FIELD?
<b>Arguments</b>	<NR3> argument is a real number from 0.0 to 8.5 in increments of 0.5. The argument sets the number of fields that the instrument waits before rearming the video trigger.
<b>Examples</b>	TRIGGER:A:VIDEO:HOLDOFF:FIELD4.5 sets the instrument to wait 4.5 video fields before rearming the trigger.  TRIGGER:A:VIDEO:HOLDOFF:FIELD? might return :TRIGGER:A:VIDEO:HOLDOFF:FIELD 5, indicating that the instrument is set to wait 5 video fields before rearming the trigger.

## TRIGger:A:VIDeo:LINE

DPO7000 Series only: This command sets or queries the video line number on which the instrument triggers. This command is equivalent to selecting Video Setup from the Trig menu, selecting Line # in the Trigger on box, and setting the line number. Use the [TRIGger:A:VIDeo:FIELD](#) command to actually trigger the instrument on the line that you specify with this command.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:LINE <NR1> TRIGger:A:VIDeo:LINE?
<b>Related Commands</b>	<a href="#">TRIGger:A:VIDeo:FIELD</a>
<b>Arguments</b>	<NR1> argument is an integer that sets the video line number on which the instrument triggers. The following table lists the valid choices, depending on the active video standard.

**Table 2-47: Video Line Numbering Ranges**

Video Standard	Line Number Range
CUSTOM	4–3000
NTSC	1–263 (odd) and 264–525 (even)
PAL	1–625

**Table 2-47: Video Line Numbering Ranges (cont.)**

<b>Video Standard</b>	<b>Line Number Range</b>
SECAM	1–625
HD480P60	1–520
HD720P60	1–750
HD1080150	1–1125
HD1080160	1–1125
HD1080P24	1–1125
HD1080P25	1–1125
HD1080SF24	1–1125

**Examples** TRIGGER:A:VIDEO:LINE23 sets the instrument to trigger on the line 23.  
 TRIGGER:A:VIDEO:LINE? might return :TRIGGER:A:VIDEO:LINE 10, indicating that the instrument is set to trigger on line 10.

## TRIGger:A:VIDeo:POLarity

DPO7000 Series only: This command sets or queries the polarity of the A video trigger.

**Group** Trigger

**Syntax** TRIGger:A:VIDeo:POLarity {INVERTed|NORMa1}  
 TRIGger:A:VIDeo:POLarity?

**Arguments** INVERTed argument sets the instrument to trigger on a positive video sync pulse.  
 NORMa1 argument sets the instrument to trigger on a negative video sync pulse.

**Examples** TRIGGER:A:VIDEO:POLARITYNORMAL sets the instrument to trigger on a negative video pulse.  
 TRIGGER:A:VIDEO:POLARITY? might return :TRIGGER:A:VIDEO:POLARITY INVERTED, indicating that the instrument is set to trigger on a positive video sync pulse.

## TRIGger:A:VIDeo:SCAN

DPO7000 Series only: This command sets or queries the video trigger horizontal line scan rate. This command is for compatibility with earlier instruments.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:SCAN {RATE1 RATE2 RATE3 RATE4 RATE5} TRIGger:A:VIDeo:SCAN?
<b>Arguments</b>	<p>RATE1 argument sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.</p> <p>RATE2 argument sets the range of the video line scan rate to 20 kHz through 25 kHz.</p> <p>RATE3 argument sets the range of the video line scan rate to 25 kHz through 35 kHz.</p> <p>RATE4 argument sets the range of the video line scan rate to 35 kHz through 50 kHz.</p> <p>RATE5 argument sets the range of the video line scan rate to 50 kHz through 65 kHz.</p>
<b>Examples</b>	<p>TRIGGER:A:VIDEO:SCANRATE1 sets the scan rate of the A video trigger to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).</p> <p>TRIGGER:A:VIDEO:SCAN? might return :TRIGGER:A:VIDEO:SCAN RATE2, indicating that the video line rate for the A trigger is set to Rate 2, which is 20 kHz to 23 kHz.</p>

## TRIGger:A:VIDeo:SOUrce

DPO7000 Series only: This command sets or queries the source for the A video trigger. This command is equivalent to selecting Video Setup from the Trig menu and selecting a channel from the Source drop-down menu.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:SOUrce CH<x> TRIGger:A:VIDeo:SOUrce?

- Arguments** CH<x> argument specifies one of the input channels of the instrument as the A video trigger. The value of x ranges from 1 through 4.
- Examples** TRIGGER:A:VIDEO:SOURCECH1 sets the source for A video trigger to Channel 1.
- TRIGGER:A:VIDEO:SOURCE? might return :TRIGGER:A:VIDEO:SOURCE CH2, indicating that the source for the A video trigger is set to Channel 2.

## TRIGger:A:VIDeo:STANdard

DPO7000 Series only: This command sets or queries the video standard.

- Group** Trigger
- Syntax** TRIGger:A:VIDeo:STANdard  
 {BILevelcustom|TRILevelcustom|NTSC|PAL|  
 SECAM|HD480P60|HD576P50|HD875I60|HD720P30|HD720P50|  
 HD720P60|HD1080I50|HD1080I60|HD1080P24|  
 HD1080P25|HD1080P30|HD1080P50|HD1080P60| HD1080SF24}  
 TRIGger:A:VIDeo:STANdard?
- Arguments** BILevelcustom argument sets the instrument to use custom video parameters that you set with the [TRIGger:A:VIDeo:CUSTom:SYNcInterval](#) command.
- TRILevelcustom argument sets the instrument to use custom video horizontal scan rate parameters that you set with the [TRIGger:A:VIDeo:CUSTom:LINEPeriod](#) command.
- NTSC argument sets the instrument to trigger on video signals that meet the NTSC 525/60/2:1 standard (a line rate of 525 lines per frame and a field rate of 60 Hz).
- PAL argument sets the instrument to trigger on video signals that meet the NTSC 625/50/2:1 standard (a line rate of 625 lines per frame and a field rate of 50 Hz).
- SECAM argument sets the instrument to trigger on video signals that meet the SECAM standard.
- HD480P60 argument sets the instrument to trigger on the HDTV 480/60 progressive format.
- HD576P50 argument sets the instrument to trigger on the HDTV 576/50 progressive format.
- HD875I60 argument sets the instrument to trigger on the HDTV 875/60 format.
- HD720P30 argument sets the instrument to trigger on the HDTV 720/30 progressive format.

HD720P50 argument sets the instrument to trigger on the HDTV 720/50 progressive format.

HD720P60 argument sets the instrument to trigger on the HDTV 720/60 progressive format.

HD1080I50 argument sets the instrument to trigger on HDTV 1080/50 interlaced format.

HD1080I60 argument sets the instrument to trigger on HDTV 1080/60 interlaced format.

HD1080P24 argument sets the instrument to trigger on HDTV 1080/24 progressive format.

HD1080P25 argument sets the instrument to trigger on HDTV 1080/25 progressive format.

HD1080P30 argument sets the instrument to trigger on HDTV 1080/30 progressive format.

HD1080P50 argument sets the instrument to trigger on HDTV 1080/50 progressive format.

HD1080P60 argument sets the instrument to trigger on HDTV 1080/60 progressive format.

HD1080SF24 argument sets the instrument to trigger on HDTV 1080/24 segmented frame format.

### Examples

TRIGGER:A:VIDEO:STANDARDNTSC sets the instrument to trigger on NTSC-standard video signals.

TRIGGER:A:VIDEO:STANDARD? might return :TRIGGER:A:VIDEO:STANDARDNTSC, indicating that the standard for the A video trigger is set to NTSC.

## TRIGger:AUXLevel

This command sets or queries the auxiliary (Aux) level for the edge trigger. This command supersedes any other argument that sets the auxiliary trigger level.

**Group** Trigger

**Syntax** TRIGger:AUXLevel {<NR3>|ECL|TTL}  
TRIGger:AUXLevel?

**Related Commands** TRIGger:{A|B}:EDGE:SLOpe, TRIGger:{A|B}:EDGE:SOURce

**Arguments** <NR3> is the auxiliary level in volts.

ECL specifies a preset ECL level.

TTL specifies a preset TTL level.

**Examples** TRIGGER:AUXlevel TTL This command sets the auxiliary level for the edge trigger to TTL.

TRIGGER:AUXlevel? might return :TRIGGER:AUXLEVEL TTL, indicating that the auxiliary level for the edge trigger is set to TTL.

## TRIGGER:B (No Query Form)

This command sets the B trigger level to 50% of minimum and maximum. The query form of this command returns the B trigger parameters. This command is similar to selecting B Event (Delayed) Trigger Setup from the Trig menu and then viewing the current setups.

**Group** Trigger

**Syntax** TRIGGER:B SETLevel

**Related Commands** [TRIGGER:A](#) [TRIGGER:B:STATE](#)

**Arguments** SETLevel sets the B trigger level to 50% of MIN and MAX.

**Examples** TRIGGER:BSETLEVEL sets the B trigger level to 50% of MIN and MAX.

TRIGGER:B? might return the following B trigger parameters:

```
:TRIGGER:B:STATE 0;TYPE EDGE; LEVEL -220.0000E-3;BY
TIME;EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC; :TRIGGER:B:TIME
16.0000E-9;EVENTS:COUNT 2.
```

## TRIGGER:B:BY

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger. This is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, and then choosing Trig After Time or Trig on nth event.

**NOTE.** *The traditional Runs After functionality is now served by the Horizontal Delay function. For details, see the [HORizontal\[:MAIn\]:DELay:MODE](#) and [HORizontal\[:MAIn\]:DELay:TIME](#) commands.*

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:B:BY {EVENTS TIME} TRIGger:B:BY?
<b>Related Commands</b>	<a href="#">TRIGger:B:EVENTS:COUNT</a> , <a href="#">TRIGger:B:TIME</a> , <a href="#">HORizontal[:MAIn]:DELay:MODE</a> , <a href="#">HORizontal[:MAIn]:DELay:TIME</a> <a href="#">TRIGger:B:STATE</a>
<b>Arguments</b>	<p>EVENTS sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by <code>TRIGger:B:EVENTS:COUNT</code>.</p> <p>TIME sets the B trigger to occur a set time after the A trigger event. The time period is specified by <code>TRIGger:B:TIME</code>.</p>
<b>Examples</b>	<p><code>TRIGGER:B:BYTIME</code> sets the B trigger to occur at a set time after the A trigger event.</p> <p><code>TRIGGER:B:BY?</code> might return <code>:TRIGGER:B:BY EVENTS</code>, indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.</p>

## TRIGger:B:EVENTS? (Query Only)

This query-only command returns the current B trigger events parameter. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, choosing Trig on nth event, and viewing the Trig Event setting.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:B:EVENTS?
<b>Related Commands</b>	<a href="#">TRIGger:B:EVENTS:COUNT</a> , <a href="#">TRIGger:B:STATE</a>

**Examples** TRIGGER:B:EVENTS? might return  
 :TRIGGER:B:EVENTS:COUNT 2, indicating that the two events must occur before the B trigger occurs.

## TRIGger:B:EVENTS:COUNT

This command sets or queries the number of events that must occur before the B trigger (when TRIG:DElay:BY is set to EVENTS). This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, choosing Trig on nth event, and setting the desired Trig Event value.

**Group** Trigger

**Syntax** TRIGger:B:EVENTS:COUNT <NR1>  
 TRIGger:B:EVENTS:COUNT?

**Related Commands** [TRIGger:B:EVENTS?](#), [TRIGger:B:STATE](#)

**Arguments** <NR1> is the number of B trigger events, which can range from 1 to 10,000,000.

**Examples** TRIGGER:B:EVENTS:COUNT4 sets the number of B trigger events to four.  
 TRIGGER:B:EVENTS:COUNT? might return :TRIGGER:B:EVENTS:COUNT 2, indicating that two events must occur after the A trigger before the B trigger can occur.

## TRIGger:B:RESET:SOUrce

This command sets or queries the trigger source for the A→B sequential trigger reset feature, except for the Timeout trigger type.

**Group** Trigger

**Syntax** TRIGger:B:RESET:SOUrce {CH<x>|AUXiliary}  
 TRIGger:B:RESET:SOUrce?

**Related Commands** [TRIGger:B:RESET:TYPE](#), [TRIGger:B:RESET:THReshold](#)



**Arguments** CH<x> specifies one of the input channels as the reset source. Input channels are specified by x and can be 1, 2, 3, or 4..

AUXiliary specifies an external trigger (using the Auxiliary Trigger Input connector) as the reset source.

**Examples** TRIGGER:B:RESET:SOURCE CH4 sets Channel 4 as the input source for the trigger reset.

TRIGGER:B:RESET:SOURCE? might return :TRIGGER:B:RESET:SOURCE CH1, indicating that the current input source for the trigger reset is Channel 1.

## TRIGger:B:RESET:STATE

This command sets or queries the trigger reset state for the A→B sequential State trigger reset feature.

**Group** Trigger

**Syntax** TRIGger:B:RESET:STATE {HIGH|LOW}  
TRIGger:B:RESET:STATE?

**Related Commands** [TRIGger:B:RESET:TYPE](#), [TRIGger:B:RESET:SOURce](#), [TRIGger:B:RESET:THReshold](#)

**Arguments** HIGH sets the condition for the state sequential trigger reset to high.

LOW sets the condition for the state sequential trigger reset to low.

**Examples** TRIGGER:B:RESET:STATE LOW sets the condition for the state sequential trigger reset to Low.

TRIGGER:B:RESET:STATE? might return :TRIGGER:B:RESET:STATE HIGH, indicating that High is the condition for the state sequential trigger reset.

## TRIGger:B:RESET:THReshold

This command sets or queries the trigger threshold for the A→B sequential trigger reset, except for the Timeout reset type.

**Group** Trigger

**Syntax** TRIGger:B:RESET:THResho1d {ECL|TTL|<NR3>}  
 TRIGger:B:RESET:THResho1d?

**Related Commands** [TRIGger:B:RESET:STATE](#)

**Arguments** ECL sets the reset threshold to ECL voltage levels.  
 TTL sets the reset threshold to TTL voltage levels.  
 <NR3> sets the reset threshold in volts.

**Examples** TRIGGER:B:RESET:THRESHOLD TTL sets the trigger reset threshold level to TTL.  
 TRIGGER:B:RESET:THRESHOLD? might return  
 :TRIGGER:B:RESET:THRESHOLD TTL, indicating that the trigger reset threshold level is set to TTL.

## TRIGger:B:RESET:TIMEOut

This command sets or queries the reset timer for a sequential timeout trigger reset. For example, if the timeout is set to 1 s, this additional reset time begins following an A trigger event and runs sequentially with any B Trigger Delay. If a B trigger event is not found before it expires, the instrument begins looking for a new A event trigger sequence.

**Group** Trigger

**Syntax** TRIGger:B:RESET:TIMEOut <NR3>  
 TRIGger:B:RESET:TIMEOut?

**Related Commands** [TRIGger:B:RESET:TYPe](#)

**Arguments** <NR3> specifies the reset time.

**Examples** TRIGGER:B:RESET:TIMEOUT 1.0E-6 sets the trigger reset timeout to 1 s.  
 TRIGGER:B:RESET:TIMEOUT? might return :TRIGGER:B:RESET:TIMEOUT 2.0000E-09, indicating that the trigger reset timeout is 2 ns.

## TRIGger:B:RESET:TRANSition

This command sets or queries the type of threshold required for a Transition trigger reset.

**Group** Trigger

**Syntax** TRIGger:B:RESET:TRANSition {RISe|FALL}  
TRIGger:B:RESET:TRANSition?

**Related Commands** [TRIGger:B:RESET:SOUrce](#), [TRIGger:B:RESET:THReshold](#), [TRIGger:B:RESET:TYPe](#)

**Arguments** RISe indicates that a reset occurs when the trigger rises above the threshold level.  
FALL indicates that a reset occurs when the trigger falls below the threshold level.

**Examples** TRIGGER:B:RESET:TRANSITION RISE allows the reset to occur if the trigger rises above the threshold level.  
  
TRIGGER:B:RESET:TRANSITION? might return  
:TRIGGER:B:RESET:TRANSITION FALL, indicating that the reset will occur if the trigger falls below the threshold level.

## TRIGger:B:RESET:TYPe

This command sets or queries the type of A→B sequential trigger reset. If the B trigger reset is active, the reset criteria are part of the B triggering sequence. If the reset conditions defined by the reset type are not met, the instrument must start over searching for a new occurrence of the A event.

You must identify a trigger Source and Threshold for each reset type, except for the Timeout trigger type.

---

**NOTE.** *If a reset condition occurs, the reset criteria itself is reset and must start over.*

---

**Group** Trigger

**Syntax** TRIGger:B:RESET:TYPe {NONE|TIMEOut|STATE|TRANSition}  
TRIGger:B:RESET:TYPe?

**Arguments** NONE defeats the trigger reset feature.

TIMEOUT initiates a reset if the timeout conditions specified by TRIGGER:B:RESET:TIMEOUT are met.

STATE initiates a reset if the state conditions specified by TRIGGER: B:RESET:SOURCE, TRIGGER:B:RESET:STATE, and TRIGGER:B:RESET:THRESHOLD are met.

TRANSITION initiates a reset if the transition conditions specified by TRIGGER:B:RESET:SOURCE, TRIGGER:B:RESET:TRANSITION, and TRIGGER:B:RESET:THRESHOLD are met.

**Examples** TRIGGER:B:RESET:TYPE NONE deactivates the sequential trigger reset.

TRIGGER:B:RESET:TYPE? This query might return :TRIGGER:B:RESET:TYPE TIMEOUT, indicating that the sequential trigger reset is active following a timeout.

## TRIGGER:B:SCAN:ADVANCEafter

This command sets or queries the B-Event Scan advance after value. Advance After advances the B-Event value after the specified number of acquisitions have occurred. The B-Events value is updated in a sequential, random, or toggle mode.

**Group** Trigger

**Syntax** TRIGGER:B:SCAN:ADVANCEafter <NR1>  
TRIGGER:B:SCAN:ADVANCEafter?

**Arguments** <NR1> is the value that advances the B-Event after the given number of acquisitions have occurred. The default value is 1 and ranges from 1 to 2e9 (2000000000).

**Examples** TRIGGER:B:SCAN:ADVANCEAFTER 1 with B-Event scan enabled, the B-Event advances after each acquisition.

TRIGGER:B:SCAN:ADVANCEAFTER? might return :TRIGGER:B:SCAN:ADVANCEAFTER:2, indicating that with B-Event scan enabled, the B-Event advances after two acquisitions.

## TRIGGER:B:SCAN:ENABle

This command sets or queries if the B-Event Scan is on or off. The B-Events value is updated in a sequential, random, or toggle Advance mode.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:SCAN:ENABle TRIGger:B:SCAN:ENABle {ON OFF} TRIGger:B:SCAN:ENABle?</pre>
<b>Arguments</b>	<p>ON with A-&gt;B Sequence mode set to Trig on nth Event, the B-Event count is determined by the other B Scan triggering parameters such as Start Event value, End Event value, Advance After value, and the Advance mode.</p> <p>OFF with A-&gt;B Sequence mode set to Trig on nth Event, behaves in a conventional way.</p>
<b>Examples</b>	<p>TRIGGER:B:SCAN:ENABLE ON enables the B-Event Scan option.</p> <p>TRIGGER:B:SCAN:ENABLE? might return :TRIGGER:B:SCAN:ENABLE:OFF, indicating that the B-Event Scan option is not enabled.</p>

## TRIGger:B:SCAN:ENDevent

This command sets or queries the end event value of B-Event Scan. The end event value is greater than or equal to the start event value.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:SCAN:ENDevent &lt;NR1&gt; TRIGger:B:SCAN:ENDevent?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:B:SCAN:STARTevent</a>
<b>Arguments</b>	<NR1> is the end event value of B-Event Scan. The default value is 8 and ranges from 1 to 2e9.
<b>Examples</b>	<p>TRIGGER:B:SCAN:ENDEVENT 8 sets the end event value of B-Event Scan to 8. At this point the B-Event value is set to the start event value again.</p> <p>TRIGGER:B:SCAN:ENDEVENT? might return :TRIGGER:B:SCAN:ENDEVENT 10, indicating that the end event value of B-Event scan is set to 10.</p>

## TRIGger:B:SCAN:MODE

This command sets or queries the sequence advance mode of B-Event Scan.

**Group** Trigger

**Syntax** TRIGger:B:SCAN:MODE {SEQUENTIAL|RANDOM|TOGGLE}  
TRIGger:B:SCAN:MODE?

**Arguments** SEQUENTIAL, where the B-Event Scan is initialized at start event. This value is incremented after each set of Advance After triggers have occurred, until the B-Event exceeds the end event value. Then the B-Event count is again set to the start event value.

RANDOM, where the B-Event value is changed to a random value such that start event value  $\leq$  BEvent  $\leq$  end event value), and after Advance After triggers have occurred.

TOGGLE, where the B-Event count is first set to the start event value and then changed to the end event value after additional Advance After triggers have occurred, and then B-Event is set back to the start event value.

**Examples** TRIGGER:B:SCAN:MODE SEQUENTIAL sets the advance mode of the B-Event Scan to sequential.

TRIGGER:B:SCAN:MODE? might return :TRIGGER:B:SCAN:MODE SEQUENTIAL, indicating that the advance mode of the B-Event Scan is set to sequential.

## TRIGger:B:SCAN:STARTEvent

This command sets or queries the start event value of B-Event Scan.

**Group** Trigger

**Syntax** TRIGger:B:SCAN:STARTEvent <NR1>  
TRIGger:B:SCAN:STARTEvent?

**Related Commands** [TRIGger:B:SCAN:ENDEvent](#)

**Arguments** <NR1> is the start event value of B-Event Scan. The default value is 1 and ranges from 1 to 2e9.

**Examples** TRIGGER:B:SCAN:STARTEVENT 1 initializes the start event value of B-Event Scan at 1.

TRIGGER:B:SCAN:STARTEVENT? might return :TRIGGER:B:SCAN:STARTEVENT 2, indicating that the start event value of B-Event scan is initialized at 2.

## TRIGger:B:STATE

This command sets or queries the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.

**Group** Trigger

**Syntax** TRIGger:B:STATE {ON|OFF|<NR1>}  
TRIGger:B:STATE?

**Related Commands** [TRIGger:A:MODE](#)

**Arguments** ON indicates that the B trigger is active and causes trigger events in conjunction with the A trigger.

OFF indicates that only the A trigger causes trigger events.

<NR1>

A 0 turns off the B trigger; any other value activates the B trigger.

**Examples** TRIGGER:B:STATEON sets the B trigger to active, making it capable of causing trigger events.

TRIGGER:B:STATE? might return :TRIGGER:B:STATE 0, indicating that the B trigger is inactive and that only the A trigger causes trigger events.

## TRIGger:B:TIME

This command sets or queries B trigger delay time. The B Trigger time applies only if TRIGger:B:BY is set to TIME. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, choosing the A®B Seq tab, and setting Trig Delay.

**Group** Trigger

**Syntax** TRIGGER:B:TIME <NR3>  
TRIGGER:B:TIME?

**Related Commands** TRIGGER:B:BY, TRIGGER:B:EVENTS:COUNT, TRIGGER:B:STATE

**Arguments** <NR3> is the B trigger delay time in seconds.

**Examples** TRIGGER:B:TIME4E-6 sets the B trigger delay time to 4  $\mu$ s.  
TRIGGER:B:TIME? might return :TRIGGER:B:TIME 16.0000E-9, indicating that the B trigger time is set to 16 ns.

## TRIGGER:ENHANCED

This command sets or queries the state of trigger position enhancement. When on, the instrument improves the trigger positioning to more closely match the acquired data. This is equivalent to selecting Enhanced Triggering from the Trigger Mode menu.

**Group** Trigger

**Syntax** TRIGGER:ENHANCED {<NR1>|OFF|ON}  
TRIGGER:ENHANCED?

**Arguments** <NR1> = 0 disables trigger position enhancement, any other value enables trigger position enhancement.  
OFF disables trigger position enhancement.  
ON enables trigger position enhancement.

**Examples** TRIGGER:ENHANCED ON enables trigger position enhancement.  
TRIGGER:ENHANCED? might return :TRIGGER:ENHANCED 0, indicating that trigger position enhancement is disabled.

## TRIGGER:LVLsrcpreference

This command sets or queries the dependent source/level trigger feature. SRCIndependent is the default value.



<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:LVLSrcpreference {SRCDependent SRCIndependent} TRIGger:LVLSrcpreference?
<b>Related Commands</b>	TRIGger: {A B}:LEVel:CH<x>,
<b>Arguments</b>	SRCDependent sets the level of all trigger sources to the value you currently select, regardless of the last value selected.  SRCIndependent sets each trigger source to the level you are currently selecting.
<b>Examples</b>	TRIGger:LVLSRCPREFERENCE SRCIINDEPENDENT allows you to set the level of each source to the value you are currently selecting.  TRIGGER:LVLSRCPREFERENCE? might return :TRIGGER:LVLSRCPREFERENCE SRCDEPENDENT, indicating that the level of all sources changes to the value you currently select.

## TRIGger:MULTIscope

This command sets or queries the state of MultiScope triggering, either ENABLE or DISABLE.

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTIscope {DISable ENable} TRIGger:MULTIscope?
<b>Arguments</b>	DISable sets the MultiScope triggering state to disabled.  ENable sets the MultiScope triggering state to enabled.
<b>Examples</b>	TRIGGER:MULTISCOPE ENABLE allows you to use the MultiScope triggering.  TRIGGER:MULTISCOPE? might return :TRIGGER:MULTISCOPE DISABLE, indicating that the MultiScope triggering is disabled.

## TRIGger:MULTiscope:ALIGN (No Query Form)

This command starts the TekLink trigger alignment procedure on all oscilloscopes which are participating in MultiScope triggering. For this command to have any effect, the instrument must be participating in MultiScope triggering. Other instruments on the TekLink network are not affected.

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTiscope:ALIGN
<b>Examples</b>	TRIGGER:MULTISCOPE:ALIGN starts the TekLink trigger alignment procedure on all the oscilloscopes that are participating in MultiScope triggering.

## TRIGger:MULTiscope:ALIGN:COMPLETED? (Query Only)

This command returns whether the oscilloscope has successfully completed the MultiScope trigger align procedure.

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTiscope:ALIGN:COMPLETED?
<b>Returns</b>	TRUE when the MultiScope trigger align procedure has successfully completed. FALSE when the MultiScope trigger align procedure fails.
<b>Examples</b>	TRIGGER:MULTISCOPE:ALIGN:COMPLETED? might return :TRIGGER:MULTISCOPE:ALIGN:COMPLETED TRUE, indicating that the MultiScope trigger align procedure has successfully completed.

## TRIGger:MULTiscope:ALIGN:DESKEW? (Query Only)

This command sets or queries the deskew time, in nanoseconds, for horizontal positioning of the waveform .

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTiscope:ALIGN:DESKEW?

**Examples** TRIGGER:MULTISCOPE:ALIGN:DESKEW? might return  
:TRIGGER:MULTISCOPE:ALIGN:DESKEW 2.50000E-9, indicating  
that the deskew time is 2.5 ns.

## TRIGger:MULTiscope:ALIGN:MRTTime

This command returns or sends the maximum round trip time to each of the instruments in a calibration cycle.

**Group** TekLink

**Syntax** TRIGger:MULTiscope:ALIGN:MRTTime  
TRIGger:MULTiscope:ALIGN:MRTTime?

**Examples** TRIGGER:MULTISCOPE:ALIGN:MRTTIME? might return  
:TRIGGER:MULTISCOPE:ALIGN:MRTTIME 20.000E-6, indicating  
that the round trip time is 20 us.

## TRIGger:MULTiscope:ALIGN:SETSTATE

This command causes the instrument to measure the round trip time between sending a trigger out the TekLink port and receiving the trigger pulse back from the port.

**Group** TekLink

**Syntax** TRIGger:MULTiscope:ALIGN:SETSTATE {START|STOP|WAIT|DONE}  
TRIGger:MULTiscope:ALIGN:SETSTATE?

**Related Commands** [TRIGger:MULTiscope:ALIGN:VALue?](#)

**Arguments** START begins calibration.  
STOP ends calibration.  
WAIT postpones calibration.  
DONE indicates that the calibration is complete.

- Examples** TRIGGER:MULTISCOPE:ALIGN:SETSTATE Start causes the instrument to measure the round trip time between sending a trigger out the TekLink port and receiving the trigger pulse back from the port.
- TRIGGER:MULTISCOPE:ALIGN:SETSTATE? might return  
:TRIGGER:MULTISCOPE:ALIGN:SETSTATE DONE, indicating that the calibration is complete.

## TRIGger:MULTiScope:ALIGN:VALue? (Query Only)

This command returns the time interval result from the TRIGGER:MULTISCOPE:ALIGN command. Round trip time, trigger out to trigger in.

**Group** TekLink

**Syntax** TRIGger:MULTiScope:ALIGN:VALue?

- Examples** TRIGGER:MULTISCOPE:ALIGN:VALUE? might return  
:TRIGGER:MULTISCOPE:ALIGN:VALUE 33.000E-6, indicating that the round trip trigger time is 33  $\mu$ s.

## TRIGger:MULTiScope:DELay? (Query Only)

This command queries the delay time, in nanoseconds, for trigger out of the TekLink trigger out line.

**Group** TekLink

**Syntax** TRIGger:MULTiScope:DELay?

- Examples** TRIGGER:MULTISCOPE:DELAY? might return :TRIGGER:MULTISCOPE:DELAY 3.5000E-9, indicating that the set TekLink trigger delay time is 3.5 ns.

## TRIGger:MULTiScope:LOGic

This command sets or queries the TekLink trigger configuration when the TekLink network connection is HUB.

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTiScope:LOGic {AND OR} TRIGger:MULTiScope:LOGic?
<b>Arguments</b>	AND sets the hub and all participating instruments to AND triggering configuration. OR sets the hub and all participating instruments to OR triggering configuration.
<b>Examples</b>	TRIGGER:MULTISCOPE:LOGIC AND sets the hub and all participating instruments to AND triggering configuration.  TRIGGER:MULTISCOPE:LOGIC? might return :TRIGGER:MULTISCOPE:LOGIC OR, indicating that the all participating instruments are set to OR triggering configuration.

## TRIGger:MULTiScope:ROLe

This command sets or queries the role of the instrument in the MultiScope trigger network. PRODUCER means that the instrument will both contribute and use the MultiScope trigger. If there is no hub, this makes the local instrument the master. CONSUMER means that the trigger in from the MultiScope network is used (rather than an internal trigger). This applies to a network with or without a hub.

<b>Group</b>	TekLink
<b>Syntax</b>	TRIGger:MULTiScope:ROLe {PRODucer CONSUMER NONE} TRIGger:MULTiScope:ROLe?
<b>Arguments</b>	PRODucer sets the instrument to both contribute and use the MultiScope trigger. CONSUMER sets the instrument to use the MultiScope trigger. NONE takes the instrument off the MultiScope trigger network.
<b>Examples</b>	TRIGGER:MULTISCOPE:ROLE Producer sets the instrument to supply triggers to the MultiScope trigger network.  TRIGGER:MULTISCOPE:ROLE? might return :TRIGGER:MULTISCOPE:ROLE CONSUMER, indicating that the instrument consumes triggers.

## TRIGger:QUALification:BUS:FORMat

This command sets or queries the format of the bit pattern for the bus qualification. This command is available only when the Trigger Type is set to Glitch, Runt, Setup/Hold, Timeout, Transition, Width, or Window and the trigger qualification is set to Occurs And.

**Group** Trigger

**Syntax** TRIGger:QUALification:BUS:FORMat {BINary|HEX|SYMBOLic}  
TRIGger:QUALification:BUS:FORMat?

**Related Commands** [TRIGger:QUALification:BUS:SOURce](#), [TRIGger:QUALification:BUS:VALue](#)

**Arguments** BINary specifies the binary pattern format.  
HEX specifies the hexadecimal pattern format.  
SYMBOLic specifies the symbolic pattern format.

**Examples** TRIGGER:QUALIFICATION:BUS:FORMAT HEX sets the bit pattern format to hexadecimal.  
TRIGGER:QUALIFICATION:BUS:FORMAT? might return :TRIGGER:QUALIFICATION:BUS:FORMAT BINARY indicating that the bit pattern format to binary.

## TRIGger:QUALification:BUS:SOURce

This command sets or queries the source for the bus qualification. This command is available only when the Trigger Type is set to Glitch, Runt, Setup/Hold, Timeout, Transition, Width, or Window and the trigger qualification is set to Occurs And.

**Group** Trigger

**Syntax** TRIGger:QUALification:BUS:SOURce {B<x>|CH<x>}  
TRIGger:QUALification:BUS:SOURce?

**Related Commands** [TRIGger:QUALification:BUS:FORMat](#), [TRIGger:QUALification:BUS:VALue](#)

<b>Arguments</b>	B<x> specifies the bus source. x can be 0 through 16. B0 is the same as All in the user interface.  CH<x> are always don't cares. x can be 1, 2, 3, or 4.
<b>Examples</b>	TRIGGER:QUALIFICATION:BUS:SOURCE B1 sets the source for bus qualification to B1.  TRIGGER:QUALIFICATION:BUS:SOURCE? might return :TRIGGER:QUALIFICATION:BUS:SOURCE B5 indicating that the source for bus qualification is set to B5.

## TRIGger:QUALification:BUS:VALue

This command sets or queries the bit pattern for the bus qualification. This command is available only when the Trigger Type is set to Glitch, Runt, Setup/Hold, Timeout, Transition, Width, or Window and the trigger qualification is set to Occurs And. The pattern assigned is in the order D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 Ch1 Ch2 Ch3 Ch4.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:QUALification:BUS:VALue <QString> TRIGger:QUALification:BUS:VALue?
<b>Related Commands</b>	<a href="#">TRIGger:QUALification:BUS:FORMat</a> , <a href="#">TRIGger:QUALification:BUS:SOURce</a>
<b>Arguments</b>	<QString> sets the bus value.
<b>Examples</b>	TRIGGER:QUALIFICATION:BUS:VALUE "1 XXXX" sets the bus value to 1xxx (a logic high on D0 and CH1 through CH4 to "don't care" value).  TRIGGER:QUALIFICATION:BUS:VALUE? might return :TRIGGER:QUALIFICATION:BUS:VALUE "01XXXX" indicating that the value for bus qualification is set to 01XXXX (a logic low is expected on D1, logic high on D0, and a "don't care" value on CH1 through CH4).

## TRIGger:SENSITivity

This command set or queries the state of trigger sensitivity. When on, the instrument calculates a rolling average for trigger position enhancement to suppress trigger jitter in noisy signals. This is equivalent to selecting Higher

Sensitivity for Noisy Signals from the Trigger Mode menu. This is available only when [TRIGger:ENHanced](#) or the equivalent check box control is also on.

**Group** Trigger

**Syntax** TRIGger:SENSITivity {<NR1>|OFF|ON}  
TRIGger:SENSITivity?

**Related Commands** [TRIGger:ENHanced](#)

**Arguments** <NR1> = 0 disables rolling averaging, any other value enables rolling averaging for trigger position enhancement.

OFF disables rolling averaging for trigger position enhancement.

ON enables rolling averaging for trigger position enhancement.

**Examples** TRIGGER:SENSITIVITY ON enables the trigger sensitivity feature when [TRIGger:ENHanced](#) is enabled.

TRIGGER:SENSITIVITY? might return:TRIGGER:SENSITIVITY 1, indicating theTRIGGER:SENSITIVITY feature is enabled.

## TRIGger:STATE? (Query Only)

This query-only command returns the current state of the triggering system. This command is equivalent to viewing the trigger status LEDs on the instrument front panel.

**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.

DPO indicates that the instrument is in DPO mode.



PARTIAL indicates that the A trigger has occurred and the instrument is waiting for the B trigger to occur.

READY indicates that all pretrigger information has been 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)

This query-only command tests (self-test) the GPIB interface and returns a 0.

**Group** Miscellaneous

**Syntax** \*TST?

**Examples** \*TST? always returns 0.

## UNLock (No Query Form)

This command (no query form) unlocks the front panel. The command is equivalent to LOCK NONE.

---

**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.

## 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). However, this command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.*

---

**Group** Miscellaneous

**Syntax** VERBose {OFF|ON|<NR1>}

**Related Commands** [HEADer](#), [\\*LRN?](#), [SET?](#)

**Arguments** 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.

<NR1>

A 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

**Examples** VERBOSEON 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.

## \*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 returns WFMOutpre? and CURVe? data for the waveform as specified by the [DATA:SOURce](#) command. This command is equivalent to sending both WFMOutpre? and CURVe?, with the additional provision that the response to WAVFrm? is guaranteed to provide a synchronized preamble and curve.

**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,samp1 mode"; :CURVE51,50,51,48,51,48,50,49,51,49,51,48,51,48,51,49,50,49,50,48,49,49,52,49,49,50,50,48,50,49,49,49,49,49,50,47,49,47,50,48,49,48,50,48,49,47,49,48,51,48,49,47,50,48,50,47,51,47,49,48,48,48,50,46,50,46,48,45,48,47,49,47,49,48,49,48,49,45,49,47,48,46,48,48,49,45,49,45,47,46,46,46,49,47,48,46,48,46,48,47,47,46,47,47,48,46,

48, 45, 48, 46, 47, 46, 47, 47, 46, 45, 46, 45, 47, 47, 47, 46, 46, 44, 47, 45, 45, 45, 46, 43, 46, 45, 46, 44, 46, 45, 48, 45, 45, 44, 46, 45, 46, 45, 45, 45, 45, 43, 45, 44, 46, 44, 47, 44, 44, 44, 45, 43, 44, 44, 45, 44, 46, 43, 46, 43, 45, 44, 45, 44, 45, 41, 44, 43, 45, 42, 44, 43, 44, 43, 44, 42, 43, 42, 44, 41, 44, 41, 44, 42, 43, 42, 43, 42, 45, 42, 43, 41, 42, 41, 43, 42, 44, 42, 43, 42, 43, 40, 42, 40, 42, 40, 42, 41, 43, 41, 42, 41, 42, 39, 41, 41, 42, 42, 43, 41, 42, 39, 41, 39, 41, 40, 41, 40, 41, 39, 41, 38, 41, 39, 41, 38, 42, 39, 41, 37, 40, 38, 40, 38, 39, 38, 39, 38, 41, 39, 39, 38, 40, 37, 38, 37, 38, 38, 39, 38, 40, 36, 39, 37, 39, 37, 39, 37, 39, 36, 40, 38, 38, 35, 38, 36, 38, 37, 39, 37, 38, 37, 38, 37, 37, 36, 38, 35, 38, 36, 37, 36, 38, 36, 38, 35, 39, 34, 36, 36, 36, 35, 38, 35, 36, 34, 37, 34, 37, 35, 37, 36, 37, 35, 37, 35, 36, 34, 36, 33, 36, 35, 36, 34, 36, 33, 34, 33, 34, 34, 34, 34, 34, 36, 33, 36, 34, 36, 33, 34, 33, 34, 32, 34, 32, 34, 33, 33, 31, 33, 32, 34, 32, 33, 32, 32, 31, 33, 31, 31, 32, 33, 31, 32, 31, 32, 30, 32, 30, 32, 31, 31, 30, 31, 30, 31, 29, 31, 29, 32, 31, 31, 30, 30, 29, 30, 29, 30, 30, 31, 29, 30, 28, 30, 29, 31, 28, 30, 28, 30, 29, 29, 27, 29, 28, 29, 27, 30, 27, 30, 26, 28, 28, 29, 27, 28, 27, 28, 27, 28, 28, 27, 28, 27, 28, 26, 28, 26, 28, 26, 26, 26, 25, 27, 25, 28, 27, 28, 26, 25, 25, 26, 25, 25, 25, 25, 25, 26, 25, 27, 25, 25, 24, 25, 25, 26, 24, 26, 24, 24, 23, 25, 24, 24, 22, 25, 23, 24, 23, 24, 22, 24, 22, 23, 22, 25, 24, 24, 21, 23, 22, 22, 21, 24, 22, 22, 22, 22, 21, 22, 20, 23, 21, 21, 20.

## WAVFRMStream? (Query Only)

This query only command returns WFMOUTPRE? and CURVESTREAM? data for the waveforms specified by the DATASOURCE command. This command is similar to sending both WFMOUTPRE? and CURVESTREAM?, with the additional provision that each CURVESTREAM response to WAVFRMS? has a WFMOUTPRE response prepended to it. This helps guarantee a continuous synchronized preamble and curve.

**Group** Waveform Transfer

**Syntax** WAVFRMStream?

**Returns** WFMOUTPRE? and CURVESTREAM? query responses.

**Examples** Single data source (1 channel, record length 1000). `WAVFRMSTREAM?` might return

```
:WFMOUTPRE <WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
. . .
```

Multiple data source (record length 1000). `WAVFRMSTREAM?` might return (wf<x> denotes one of each waveform source specified by `:DATA:SOURCE WF1,WF2,WF3`)

```
:WFMOUTPRE <WF1 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WF2 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WF3 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WF1 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WF2 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
:WFMOUTPRE <WF3 WFMOUTPRE branch query>;:CURVE #41000<binary data>;<newline>
. . .
```

## WFMInpre? (Query Only)

This query-only command returns the waveform formatting specification to be applied to the next incoming CURVE command data.

**Group** Waveform Transfer

**Syntax** `WFMInpre?`

**Related Commands** [WFMOutpre?](#)

**Examples** `WFMINPRE?` might return the waveform formatting as `:WFMINPRE:BIT_NR 8;BN_FMTRI;BYT_NR 1; BYT_OR MSB;ENCDG BIN;NR_PT 500;PT_FMT`

Y; PT\_OFF 0;XINCR 2.0000E-6;XZERO 1.7536E-6; XUNIT "s";YMULT 1.0000E-3;YOFF 0.0000; YZERO 0.0000;YUNIT "v".

## WFMinpre:BIT\_Nr

This command sets or queries the number of bits per binary waveform point for the waveform, as specified by the [DATA:DESTINATION](#) command. This specification is meaningful only when [WFMinpre:ENCdg](#) is set to BIN.

**Group** Waveform Transfer

**Syntax** WFMinpre:BIT\_Nr <NR1>  
WFMinpre:BIT\_Nr?

**Related Commands** [DATA:DESTINATION](#), [WFMinpre:ENCdg](#), [WFMinpre:BYT\\_Nr](#), [WFMinpre:BIT\\_Nr](#)

**Arguments** <NR1> number of bits per data point can be 8, 16 ( RI, RP) or 32 (FP).

**Examples** WFMinpre:BIT\_Nr16 sets the number of bits per waveform point to 16, for incoming RI and RP binary format data.

WFMinpre:BIT\_Nr? might return :WFMinpre:BIT\_Nr 8, indicating that incoming RI or RP binary format data uses 8 bits per waveform point.

## WFMinpre:BN\_Fmt

This command sets or queries the format of binary data for incoming waveforms.

**Group** Waveform Transfer

**Syntax** WFMinpre:BN\_Fmt {RI|RP|FP}  
WFMinpre:BN\_Fmt?

**Related Commands** [WFMinpre:BN\\_Fmt](#)

**Arguments** RI specifies signed integer data point representation.  
RP specifies positive integer data point representation.

FP specifies single-precision binary floating point representation.

**Examples** `WFMINPRE:BN_FMTFP` specifies that incoming data will be interpreted as single-precision binary floating point numbers.

`WFMINPRE:BN_FMT?` might return `:WFMINPRE:BN_FMT RI`, indicating that the incoming data is currently interpreted as signed integers.

## WFMinpre:BYT\_Nr

This command sets or queries the binary field data width for the first ordered waveform, as specified by the [DATA:DESTINATION](#) command. This specification is only meaningful when [WFMinpre:ENCdg](#) is set to BIN and [WFMinpre:BN\\_Fmt](#) is set to either RI or RP.

**Group** Waveform Transfer

**Syntax** `WFMinpre:BYT_Nr <NR1>`  
`WFMinpre:BYT_Nr?`

**Related Commands** [DATA:DESTINATION](#), [WFMinpre:BN\\_Fmt](#), [WFMinpre:ENCdg](#),  
[WFMinpre:BIT\\_Nr](#), [WFMinpre:BYT\\_Nr](#)

**Arguments** `<NR1>` is the number of bytes per data point and can be 1, 2 (RI, RP) or 4 (FP).

**Examples** `WFMINPRE:BYT_NR1` sets the number of bytes per incoming waveform data point to 1, which is the default setting.

`WFMINPRE:BYT_NR?` might return `:WFMINPRE:BYT_NR 2` indicating that there are 2 bytes per incoming waveform data point.

## WFMinpre:BYT\_Or

This command sets or queries which byte of binary waveform data is transmitted first for incoming waveform data when data points require more than one byte. This specification only has meaning when [WFMinpre:ENCdg](#) is set to BIN and [WFMinpre:BYT\\_Nr](#) is greater than 1.

**Group** Waveform Transfer

**Syntax**      `WFMInpre:BYT_Or {LSB|MSB}`  
`WFMInpre:BYT_Or?`

**Related Commands**    [WFMInpre:ENCdg](#), [WFMInpre:BYT\\_Nr](#), [WFMOutpre:BYT\\_Or](#)

**Arguments**      `LSB` specifies that the least significant byte will be transmitted first.  
`MSB` specifies that the most significant byte will be transmitted first.

**Examples**        `WFMINPRE:BYT_ORMSB` sets the most significant incoming byte of incoming waveform data to be transmitted first.  
`WFMINPRE:BYT_OR?` might return `:WFMINPRE:BYT_OR LSB`, indicating that the least significant incoming CURVe data byte will be transmitted first.

## WFMInpre:ENCdg

This command sets or queries the type of encoding for incoming waveform data.

**Group**            Waveform Transfer

**Syntax**        `WFMInpre:ENCdg {ASCIi|BINary}`  
`WFMInpre:ENCdg?`

**Related Commands**    [WFMOutpre:ENCdg](#)

**Arguments**      `ASCIi` specifies that the incoming data is in ASCII format.  
`BINary` specifies that the incoming data is in a binary format whose further interpretation requires knowledge of `BYT_NR`, `BIT_NR`, `BN_FMT`, and `BYT_OR`.

**Examples**        `WFMINPRE:ENCDGASCIi` sets the format of incoming waveform data to ASCII format.  
`WFMINPRE:ENCDG?` might return `:WFMINPRE:ENCDG BINary`, indicating that the incoming waveform data is in binary format.

## WFMInpre:NR\_FR

This command sets or queries one frame, indicating the number of frames in the transmitted waveform record.



<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMinpre:NR_FR WFMinpre:NR_FR?
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA</a> , <a href="#">DATA:START</a> , <a href="#">DATA:STOP</a> , <a href="#">SAVE:WAVEform</a> , <a href="#">SAVE:WAVEform:FILEFormat</a> , <a href="#">WFMinpre:NR_Pt</a> , <a href="#">WFMinpre:NR_FR?</a> , <a href="#">WFMinpre:NR_Pt?</a>
<b>Examples</b>	WFMINPRE:NR_FR 4 specifying to send 4 data frames in the transmitted waveform record.  WFMINPRE:NR_FR? might return :WFMINPRE:NR_FR 1, indicating that there is 1 data frame in the transmitted waveform record.

## WFMinpre:NR\_Pt

This command sets or queries the number of data points that are in the transmitted waveform record.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMinpre:NR_Pt <NR1> WFMinpre:NR_Pt?
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA</a> , <a href="#">DATA:START</a> , <a href="#">DATA:STOP</a> , <a href="#">SAVE:WAVEform</a> , <a href="#">SAVE:WAVEform:FILEFormat</a> , <a href="#">WFMinpre:NR_FR</a> , <a href="#">WFMinpre:NR_FR?</a> , <a href="#">WFMinpre:NR_Pt?</a>
<b>Arguments</b>	<NR1> is the number of data points if WFMinpre:PT_Fmt is set to Y. It is the number of min-max pairs if WFMinpre:PT_Fmt is set to ENV.
<b>Examples</b>	WFMINPRE:NR_PT5000 specifies that 5000 data points will be transmitted.  WFMINPRE:NR_PT? might return :WFMINPRE:NR_PT 8000, indicating that there are 8000 data points in the transmitted waveform record.

## WFMinpre:PT\_Fmt

This command sets or queries the point format of the incoming waveform data. Regardless of the argument used, the scale, offset, and so on are interpreted

similarly. When ENV is used, waveform data is interpreted over the min-max pair; when Y is used, it is interpreted over a single point.

**Group** Waveform Transfer

**Syntax** WFMInpre:PT\_Fmt {ENV|Y}  
WFMInpre:PT\_Fmt?

**Related Commands** [WFMOutpre:PT\\_Fmt?](#)

**Arguments** ENV specifies that the waveform is transmitted in envelope mode as maximum and minimum point pairs. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZERO + XINcr (N-PT\_Off)$$

$$Y_{max} = YZERO + YMULT (ynmax - YOFf)$$

$$Y_{min} = YZERO + YMULT (ynmin - YOFf)$$

Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZERO + XINcr (N-PT\_Off)$$

$$Y_n = YZERO + YMULT (Yn - YOFf)$$

**Examples** WFMINPRE:PT\_FMTENV sets the incoming waveform data point format to enveloped.

WFMINPRE:PT\_FMT? might return :WFMINPRE:PT\_FMT ENV, indicating that the waveform is transmitted as maximum and minimum point pairs.

## WFMInpre:PT\_Off

This command specifies or queries the trigger point within the waveform record for the reference waveform specified by the [DATA:DESTINATION](#) command.

**Group** Waveform Transfer

**Syntax** WFMInpre:PT\_Off <NR1>  
WFMInpre:PT\_Off?

<b>Related Commands</b>	<a href="#">DATA:DESTination</a> , <a href="#">DATA:START</a> , <a href="#">WFMOutpre:PT_Off?</a>
<b>Arguments</b>	<NR1> is -(record length -1) to record length and is the position of the data point immediately following the actual trigger.<NR1> is expressed relative to DATA:START.
<b>Examples</b>	<p>WFMINPRE:PT_OFF1 specifies that the trigger point is the same as the Data:Start point.</p> <p>WFMINPRE:PT_OFF? might return :WFMINPRE:PT_OFF 0, indicating that the incoming waveform trigger point is the first point in the waveform record.</p>

## WFMinpre:WFId (No Query Form)

This command (no query form) accepts but ignores the argument. This command is provided only to allow a waveform extracted from the instrument to be easily imported.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMinpre:WFId <QString>
<b>Related Commands</b>	<a href="#">DATA:DESTination</a> , <a href="#">WFMinpre:BN_Fmt</a> , <a href="#">WFMinpre:ENCdg</a> , <a href="#">WFMOutpre:WFId?</a>
<b>Arguments</b>	<QString> must be a valid IEEE-488.2 string (but the contents are ignored).
<b>Examples</b>	WFMINPRE:WFID"CH1, DC COUPLING, 2.000V/DIV, 400.0NS/DIV, 500 POINTS, SAMPLE MODE" is a syntactically correct command.

## WFMinpre:XINcr

This command sets or queries the horizontal interval between incoming waveform points in units specified by WFMinpre:XUNit.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMinpre:XINcr <NR3> WFMinpre:XINcr?

**Related Commands** [WFMinpre:XUNit](#), [WFMOutpre:XINcr?](#)

**Arguments** <NR3> is the horizontal interval representation.

**Examples** WFMINPRE:XINCR3E-3 sets the interval between Incoming waveform points to 3 ms.

WFMINPRE:XINCR? might return :WFMINPRE:XINCR 1.0000E-3, indicating that if WFMInpre:XUNit is set to "s", there is a 1 ms interval between incoming waveform points.

## WFMinpre:XUNit

This command sets or queries the horizontal units of the incoming waveform.

**Group** Waveform Transfer

**Syntax** WFMInpre:XUNit <QString>  
WFMInpre:XUNit?

**Related Commands** [WFMOutpre:XUNit?](#)

**Arguments** <QString> contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

**Examples** WFMINPRE:XUNIT"HZ" specifies that the horizontal units for the incoming waveform are hertz.

WFMINPRE:XUNIT? might return :WFMINPRE:XUNIT "s", indicating that the horizontal units for the incoming waveform are seconds.

## WFMinpre:XZEro

This command sets or queries the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger on the incoming waveform. This value is used to compute TTOFF for the incoming waveform, and is expressed in terms of [WFMinpre:XUNit](#).

**Group** Waveform Transfer

**Syntax** WFMInpre:XZErO <NR3>  
WFMInpre:XZErO?

**Related Commands** [WFMInpre:PT\\_Off](#), [WFMInpre:XINcr](#), [WFMInpre:XUNit](#), [WFMOutpre:XZErO?](#)

**Arguments** <NR3> argument is a floating point value that ranges from –WFMInpre:XINcr to 0.

**Examples** WFMINPRE:XZErO5.7E-6 specifies that the trigger actually occurred 5.7 μs before the sample designated by [WFMInpre:PT\\_Off](#).

WFMINPRE:XZErO? might return :WFMINPRE:XZErO 7.5000E-6, indicating that the trigger occurs 7.5 μs before the sample designated by [WFMInpre:PT\\_Off](#).

## WFMInpre:YMUIt

This command sets or queries the vertical scale factor (in units/digitizing level) for the reference waveform, specified by [DATA:DESTination](#), upon a [CURVe](#) command.

**Group** Waveform Transfer

**Syntax** WFMInpre:YMUIt <NR3>  
WFMInpre:YMUIt?

**Related Commands** [DATA:DESTination](#), [WFMInpre:BYT\\_Nr](#), [WFMInpre:YUNit](#)

**Arguments** <NR3> is the vertical scale factor per digitizing level of the incoming waveform points.

**Examples** WFMINPRE:YMULT20E-3 specifies that (if WFMInpre:YUNit is “V” and WFMInpre:BYT\_Nr is 1), the vertical scale is 20 mV/digitizing level (500 mV/div).

WFMINPRE:YMULT? might return :WFMINPRE:YMULT 40.0000E-3, indicating that the vertical scale is 40 mV/digitizing level (1V/div).

## WFMInpre:YOff

This command sets or queries the vertical position of the incoming waveform in digitizing levels. Variations in this number are analogous to changing the vertical position of the waveform. For those formats in which [WFMInpre:BYT\\_Nr](#) is

important (all non-floating point formats), this command must take the location of the binary point implied by `BYT_NR` into consideration.

**Group** Waveform Transfer

**Syntax** `WFMInpre:YOFF <NR3>`  
`WFMInpre:YOFF?`

**Related Commands** [WFMInpre:BYT\\_Nr](#), [WFMInpre:YMULT](#), [WFMOutpre:YOFF?](#)

**Arguments** `<NR3>` is the vertical offset in digitizing levels.

**Examples** `WFMINPRE:YOFF50` specifies that the zero reference point for the incoming waveform is 50 digitizing levels (2 divisions) above the center of the data range.

`WFMINPRE:YOFF?` might return `:WFMINPRE:YOFF 25`, indicating the vertical position of the incoming waveform in digitizing levels.

## WFMInpre:YUNit

This command sets or queries the vertical units of the incoming waveform.

**Group** Waveform Transfer

**Syntax** `WFMInpre:YUNit <QString>`  
`WFMInpre:YUNit?`

**Related Commands** [WFMOutpre:YUNit?](#)

**Arguments** `<QString>` contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.

**Examples** `WFMINPRE:YUNIT"PA"` specifies that the vertical units for the incoming waveform are Pascal.

`WFMINPRE:YUNIT?` might return `:WFMINPRE:YUNIT "s"`, indicating the vertical units for the incoming waveform are seconds.

## WFMInpre:YZero

This command sets or queries the offset of the incoming waveform in units specified by WFMInpre:YUNit. Variations in this number are analogous to changing the vertical offset of the waveform.

**Group** Waveform Transfer

**Syntax** WFMInpre:YZero <NR3>  
WFMInpre:YZero?

**Related Commands** [WFMInpre:YUNit](#), [WFMOutpre:YZero?](#)

**Arguments** <NR3> is the offset in YUNits.

**Examples** WFMINPRE:YZERO1.5E+0 specifies that the zero reference point for the incoming waveform is 1.5 V below the center of the data range (given that WFMInpre:YUNit is set to V).

WFMINPRE:YZERO? might return :WFMINPRE:YZero 7.5000E-6, indicating that the zero reference for the incoming waveform is 7.5  $\mu$ V below the center of the data range (given that WFMInpre:YUNit is set to V).

## 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:BIT\_NR 8;BN\_FMT RI;BYT\_NR 1; BYT\_OR MSB;ENCDG  
BIN;NR\_PT 500; PT\_FMT Y;PT\_ORDER LINEAR;PT\_OFF 0; XINCR  
8.0000E-9;XZERO 4.8794E-9; XUNIT "s";YMULT -2000.0000E-3;

YOFF -4999.9995E-3;YZERO 0.0000;YUNIT "V"; WFID "Ch1, DC coupling, 2.000V/div, 400.0ns/div,500 points, Sample mode".

## 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, 16, 32 or 64.

**Examples** WFMOUTPRE:BIT\_NR16 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 single-precision binary floating point data point representation.

**Examples** WFMOUTPRE:BN\_FMTFP 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 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, 4 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 4 bytes per waveform point indicate math data; 8 bytes per waveform point indicate pixel map (DPO) data.

**Examples** WFMOUTPRE:BYT\_NR1 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.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:BYT_Or {LSB MSB} WFMOutpre:BYT_Or?
<b>Related Commands</b>	<a href="#">WFMOutpre:ENCdg</a>
<b>Arguments</b>	LSB specifies that the least significant byte will be transmitted first. MSB specifies that the most significant byte will be transmitted first.
<b>Examples</b>	WFMOUTPRE:BYT_ORMSB 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:ENCdg

This command sets and queries the type of encoding for outgoing waveforms.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:ENCdg {ASCIi BINary} WFMOutpre:ENCdg?
<b>Related Commands</b>	<a href="#">DATA:ENCdg</a> , <a href="#">WFMOutpre:BYT_Nr</a> , <a href="#">WFMOutpre:BYT_Or</a> , <a href="#">WFMOutpre:BIT_Nr</a> , <a href="#">WFMOutpre:BN_Fmt</a>
<b>Arguments</b>	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 <a href="#">WFMOutpre:BYT_Nr</a> , <a href="#">WFMOutpre:BIT_Nr</a> , <a href="#">WFMOutpre:BN_Fmt</a> and <a href="#">WFMOutpre:BYT_Or</a> .
<b>Examples</b>	WFMOUTPRE:ENCDGASCIi 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\_FR? (Query Only)

This query-only command returns the number of frames for the [DATA:SOURce](#) waveform transmitted in response to a [CURVe?](#) query.

**Group** Waveform Transfer

**Syntax** WFMOutpre:NR\_FR?

**Related Commands** [CURVe](#), [DATA](#), [DATA:START](#), [DATA:STOP](#), [SAVE:WAVEform](#), [SAVE:WAVEform:FILEFormat](#), [WFMinpre:NR\\_FR](#), [WFMinpre:NR\\_Pt](#), [WFMOutpre:NR\\_Pt?](#)

**Examples** WFMOUTPRE:NR\_FR? might return :WFMOUTPRE:NR\_FR 500, indicating that there are 500 data frames to be sent.

## 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](#), [SAVE:WAVEform:FILEFormat](#), [WFMinpre:NR\\_FR](#), [WFMinpre:NR\\_Pt](#), [WFMOutpre:NR\\_FR?](#)

**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.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:PT_Fmt?
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA:SOURce</a>
<b>Examples</b>	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.*

---

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:PT_Off?
<b>Related Commands</b>	<a href="#">DATA:SOURce</a> , <a href="#">DATA:START</a> , <a href="#">WFMOutpre:XZerO?</a>
<b>Examples</b>	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 502 (vertical) by 1000 (horizontal) point bitmap. Each point represents display intensity for that screen location. Only [CURVe?](#) query functions are allowed on Fast Acquisition waveforms.

When the WFMOutpre:PT\_OR query returns Column, this indicates that the source is a Fast Acquisition waveform (and that each of 1000 possible horizontal columns being transmitted contains 502 vertical points). When the WFMOutpre:PT\_OR? 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\_OR COL specifying that the waveform designated by the DATA:SOURce waveform is a Fast Acquisition waveform.

WFMOUTPRE:PT\_ORDER? might return :WFMOUTPRE:PT\_OR LINEAR, indicating that the source waveform is a non-Fast Acquisition waveform.

## 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> comprises 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.	"CH1-4" "Math1-3" "Ref1-4"
Coupling	A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math).	"AC coupling" "DC coupling" "GND coupling"

**Table 2-48: Waveform Suffixes (cont.)**

Field	Description	Examples
Vert Scale	A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.	"100.0 mV/div" "20.00 dB/div" "45.00 deg/div" "785.4 mrad/div" "500.0 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"

**Examples** WFMOUTPRE:WFID? might return :WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0μs/div,500 points, Hi Res mode".

## 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  $\mu$ s/point (500  $\mu$ s/div).

## WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the [DATA:SOURce](#) command.

An error is reported if the DATA:SOURce waveform does not exist.

**Group** Waveform Transfer

**Syntax** WFMOutpre:XUNit?

**Related Commands** [DATA:SOURce](#)

**Examples** WFMOUTPRE:XUNIT? might return :WFMOUTPRE:XUNIT "HZ", indicating that the horizontal units for the waveform are in Hertz.

## WFMOutpre:XZEro? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger for the waveform specified by the [DATA:SOURce](#) command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATA:SOURce waveform does not exist.

---

**NOTE.** *During steady state operation, when all control changes have settled and triggers are arriving on a regular basis, this is the only part of the preamble that changes on each acquisition.*

---

**Group** Waveform Transfer

**Syntax** WFMOutpre:XZEro?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:XUNit?](#)

**Examples**    `WFMOUtpRE: XZERO?` might return `:WFMOUtpRE: XZERO 5.6300E-9`, indicating that the trigger actually occurred 5.63 ns before the trigger sample.

## WFMOUtpre: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 in digitized levels for the waveform specified by the `DATA:SOUrce` command. For those formats in which `BYT_NR` is important (all non-floating point formats), this command must take the location of the binary point implied by `WFMOUtpre:BYT_Nr` into consideration.

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 vertical offset in units specified by [WFMOutpre:YUNit?](#) 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: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.

## ZOOM

This command resets the zoom transforms to default values for all traces or live traces. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display.

**Group** Zoom

**Syntax** ZOOM {RESET|RESETLive}  
ZOOM?

**Arguments** RESET resets the zoom transforms to default values for all traces.  
RESETLive resets the zoom transforms to default values for live traces.

**Examples** ZOOMRESET resets the zoom transforms to default values for all traces.  
ZOOM? might return :ZOOM:MODE 0;GRATICULE:SIZE 80;SPLIT EIGHTYTWO:ZOOM:SCROLL:DIRECTION STOP; LOCK 0;SPEED 1:ZOOM:ZOOM1:STATE 1; SCROLLLOCK 1;CH1:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH2:HORIZONTAL:POSITION 50.0000;SCALE 5:ZOOM:ZOOM1:CH2:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH3:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH3:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH4:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH4:VERTICAL:POSITION 0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH1:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:MATH1:VERTICAL:POSITION 0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH2:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:MATH2:VERTICAL:POSITION 0.0000; . . .

## ZOOM:GRATICULE:SIZE

This command sets or queries the size, in percent, of the Zoom (lower) graticule.

**Group** Zoom

**Syntax** ZOOM:GRATICULE:SIZE {50|80|100}  
ZOOM:GRATICULE:SIZE?

**Arguments** 50 sets the size of the Zoom graticule to 50%.  
80 set the size of the Zoom graticule to 80%.  
100 sets the size of the Zoom graticule to 100%.

**Examples** ZOOM:GRATICULE:SIZE100 sets the zoom graticule size to 100%.  
ZOOM:GRATICULE:SIZE? might return :ZOOM:GRATICULE:SIZE 100, indicating that the zoom graticule size is set to 100%.

## ZOOM:HORIZONTAL:POSITION

This command sets or queries the horizontal position around which the zoom waveform is displayed. It is freely movable within the confines of the acquired waveform (0% to 100%) and measured from left to right of the acquired waveform.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:HORIZONTAL:POSITION <NR3> ZOOM:HORIZONTAL:POSITION?
<b>Arguments</b>	<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 1× or greater.
<b>Examples</b>	ZOOM:HORIZONTAL:POSITION50 sets the horizontal position to 50.  ZOOM:HORIZONTAL:POSITION? might return :ZOOM:HORIZONTAL:POSITION 23.90000, indicating that the horizontal position is set to 23.9.

## ZOOM:HORIZONTAL:SCALE

This command sets or queries the zoom horizontal scale factor around which the zoom waveform is displayed.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:HORIZONTAL:SCALE <NR3> ZOOM:HORIZONTAL:SCALE?
<b>Arguments</b>	<NR3> is the amount of expansion in the horizontal direction in 1-2-5 increments.
<b>Examples</b>	ZOOM:HORIZONTAL:SCALE 5 sets the horizontal scale to 5.  ZOOM:HORIZONTAL:SCALE? might return :ZOOM:HORIZONTAL:SCALE 10, indicating that the horizontal scale is set to 10.

## ZOOM:MATH<x>:HORIZONTAL:POSITION

This command sets or queries the horizontal position of the specified math waveform.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:MATH<x>:HORizontal:POSITION <NR3> ZOOM:MATH<x>:HORizontal:POSITION?
<b>Arguments</b>	<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 1× or greater.
<b>Examples</b>	ZOOM:MATH1:HORIZONTAL:POSITION 50 sets the horizontal position of the specified Math 1 waveform to 50.  ZOOM:MATH1:HORIZONTAL:POSITION? might return :ZOOM:MATH1:HORIZONTAL:POSITION 23.9000, indicating that the horizontal position of the specified Math 1 waveform is set to 23.9.

## ZOOM:MATH<x>:HORizontal:SCALE

This command sets or queries the zoom horizontal scale factor of the specified math waveform.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:MATH<x>:HORizontal:SCALE <NR3> ZOOM:MATH<x>:HORizontal:SCALE?
<b>Arguments</b>	<NR3> is the amount of expansion in the horizontal direction in 1-2-5 increments.
<b>Examples</b>	ZOOM:MATH1:HORIZONTAL:SCALE 10 sets the horizontal scale factor of the specified Math 1 waveform to 10.  ZOOM:MATH1:HORIZONTAL:SCALE? might return :ZOOM:MATH1:HORIZONTAL:SCALE 5, indicating that the horizontal scale factor of the specified Math 1 waveform to 5.

## ZOOM:MATH<x>:VERTical:POSITION

This command sets or queries the vertical position of the specified math waveform.

<b>Group</b>	Zoom
--------------	------

**Syntax** ZOOM:MATH<x>:VERTICAL:POSITION <NR3>  
ZOOM:MATH<x>:VERTICAL:POSITION?

**Arguments** <NR3> is the vertical position, expressed in divisions.

**Examples** ZOOM:MATH1:VERTICAL:POSITION 5 sets the vertical l position of the specified Math 1 waveform to 5.

ZOOM:MATH1:VERTICAL:POSITION? might return  
:ZOOM:MATH1:VERTICAL:POSITION 340.0000E-3, indicating that the vertical position of the specified Math 1 waveform is set to 0.34.

## ZOOM:MATH<x>:VERTICAL:SCALE

This command sets or queries the zoom vertical scale of the specified math waveform.

**Group** Zoom

**Syntax** ZOOM:MATH<x>:VERTICAL:SCALE <NR3>  
ZOOM:MATH<x>:VERTICAL:SCALE?

**Arguments** <NR3> sets the horizontal scale factor of the specified Math 1 waveform to 10.

**Examples** ZOOM:MATH1:VERTICAL:SCALE 5 sets the vertical scale factor of the specified Math 1 waveform to 5.

ZOOM:MATH1:VERTICAL:SCALE? might return  
:ZOOM:MATH1:VERTICAL:SCALE 2.0000, indicating that the vertical scale factor of the specified Math 1 waveform is set to 2.

## ZOOM:MODE

This command turns Zoom mode on or off. The Zoom query returns the current state of Zoom mode. This command is equivalent to pressing the **ZOOM** button located on the front panel.

**Group** Zoom

**Syntax** ZOOM:MODE {ON|OFF|<NR1>}  
ZOOM:MODE

**Arguments** ON turns on Zoom mode.  
OFF turns off Zoom mode.  
<NR1> = 0 turns off Zoom mode; any other value turns on Zoom mode.

**Examples** ZOOM:MODEOFF turns off Zoom mode.  
ZOOM:MODE? might return :ZOOM:MODE 1, indicating that Zoom mode is currently turned on.

## ZOOM:REF<x>:HORIZONTAL:POSITION

This command sets or queries the horizontal position of the specified reference waveform.

**Group** Zoom

**Syntax** ZOOM:REF<x>:HORIZONTAL:POSITION <NR3>  
ZOOM:REF<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 1× or greater.

**Examples** ZOOM:REF1:HORIZONTAL:POSITION 50 sets the horizontal position of the specified Reference 1 waveform to 50.  
ZOOM:REF1:HORIZONTAL:POSITION? might return :ZOOM:REF1:HORIZONTAL:POSITION 23.9000, indicating that the horizontal position of the specified Reference 1 waveform is set to 23.9.

## ZOOM:REF<x>:HORIZONTAL:SCALE

This command sets or queries the zoom horizontal scale factor of the specified reference waveform.

**Group** Zoom

<b>Syntax</b>	<code>ZOOM:REF&lt;x&gt;:HORIZONTAL:SCALE &lt;NR3&gt;</code> <code>ZOOM:REF&lt;x&gt;:HORIZONTAL:SCALE?</code>
<b>Arguments</b>	<NR3> is the amount of expansion in the horizontal direction in 1-2-5 increments.
<b>Examples</b>	<code>ZOOM:REF1:HORIZONTAL:SCALE 10</code> sets the horizontal scale factor of the specified Reference 1 waveform to 10.  <code>ZOOM:REF1:HORIZONTAL:SCALE?</code> might return <code>:ZOOM:REF1:HORIZONTAL:SCALE 5</code> , indicating that the horizontal scale factor of the specified Reference 1 waveform to 5.

## ZOOM:REF<x>:VERTICAL:POSITION

This command sets or queries the vertical position of the specified reference waveform.

<b>Group</b>	Zoom
<b>Syntax</b>	<code>ZOOM:REF&lt;x&gt;:VERTICAL:POSITION &lt;NR3&gt;</code> <code>ZOOM:REF&lt;x&gt;:VERTICAL:POSITION?</code>
<b>Arguments</b>	<NR3> is the vertical position, expressed in divisions.
<b>Examples</b>	<code>ZOOM:REF1:VERTICAL:POSITION 5</code> sets the vertical l position of the specified Reference 1 waveform to 5.  <code>ZOOM:REF1:VERTICAL:POSITION?</code> might return <code>:ZOOM:REF1:VERTICAL:POSITION 340.0000E-3</code> , indicating that the vertical position of the specified Reference 1 waveform is set to 0.34.

## ZOOM:REF<x>:VERTICAL:SCALE

This command sets or queries the zoom vertical scale of the specified reference waveform.

<b>Group</b>	Zoom
<b>Syntax</b>	<code>ZOOM:REF&lt;x&gt;:VERTICAL:SCALE &lt;NR3&gt;</code> <code>ZOOM:REF&lt;x&gt;:VERTICAL:SCALE?</code>

**Arguments** <NR3> sets the horizontal scale factor of the specified Reference 1 waveform to 10.

**Examples** ZOOM:REF1:VERTICAL:SCALE 5 sets the vertical scale factor of the specified Reference 1 waveform to 5.

ZOOM:REF1:VERTICAL:SCALE? might return :ZOOM:REF1:VERTICAL:SCALE 2.0000, indicating that the vertical scale factor of the specified Reference 1 waveform is set to 2.

## ZOOM:SCROLL:DIRECTION

This command sets or queries the direction for automatic scrolling of zoomed waveforms.

**Group** Zoom

**Syntax** ZOOM:SCROLL:DIRECTION {FWD|FFWD|REV|FREV|STOP}  
ZOOM:SCROLL:DIRECTION?

**Arguments** FWD starts AutoScroll. The Zoom Box moves from left to right.

FFWD starts AutoScroll. Increases the scrolling speed in the forward direction.

REV starts AutoScroll. The Zoom Box moves from right to left.

FREV starts AutoScroll. Increases the scrolling speed in the reverse direction.

STOP halts AutoScroll.

**Examples** ZOOM:SCROLL:DIRECTIONFWD starts AutoScroll. The Zoom Box moves from left to right.

ZOOM:SCROLL:DIRECTION? might return :ZOOM:SCROLL:DIRECTION REVERSE, indicating that Zoom Box is moving from right to left.

## ZOOM:SCROLL:LOCK

This command sets or queries the state of Scroll Lock, which “locks” zoomed waveforms under common control.

**Group** Zoom



<b>Syntax</b>	<code>ZOOM:SCROLL:LOCK {ON OFF &lt;NR1&gt;}</code> <code>ZOOM:SCROLL:LOCK?</code>
<b>Arguments</b>	<code>ON</code> enables Scroll Lock. <code>OFF</code> disables Scroll Lock. <code>&lt;NR1&gt; = 0</code> disables Scroll Lock for all zoom windows; any other value enables Scroll Lock for all zoom windows.
<b>Examples</b>	<code>ZOOM:SCROLL:LOCK ON</code> enables Scroll Lock for zoomed waveforms identified as locked. <code>ZOOM:SCROLL:LOCK?</code> might return <code>:ZOOM:SCROLL:LOCK OFF</code> , indicating that the scroll lock function is disabled.

## ZOOM:SCROLL:SPEED

This command sets or queries the speed of automatic scrolling.

<b>Group</b>	Zoom
<b>Syntax</b>	<code>ZOOM:SCROLL:SPEED &lt;NR1&gt;</code> <code>ZOOM:SCROLL:SPEED?</code>
<b>Arguments</b>	<code>&lt;NR1&gt;</code> is a value from 1 to 10.
<b>Examples</b>	<code>ZOOM:SCROLL:SPEED2</code> will scroll zoomed waveform(s) 2 divisions per update. <code>ZOOM:SCROLL:SPEED?</code> might return <code>:ZOOM:SCROLL:SPEED 1</code> , indicating that the scroll speed is set to move zoomed waveform(s) 1 division per update.

## ZOOM:STATE

This command sets or queries the specified zoom on or off.

<b>Group</b>	Zoom
<b>Syntax</b>	<code>ZOOM:STATE {ON OFF &lt;NR1&gt;}</code> <code>ZOOM:STATE?</code>

- Arguments** ON turns zoom on or off.  
 OFF turns zoom on or off.  
 <NR1> = 0 disables the zoom; any other value enables the zoom.
- Examples** ZOOM:STATE ON turns zoom on.  
 ZOOM:STATE? might return :ZOOM:STATE 1, indicating that zoom is on.

## ZOOM:VERTical:POSition

This command sets or queries the vertical position around which the zoom waveform is displayed. It is freely movable within the confines of the acquired waveform (0% to 100%) and measured from left to right of the acquired waveform.

- Group** Zoom
- Syntax** ZOOM:VERTical:POSition <NR3>  
 ZOOM:VERTical: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 1× or greater.
- Examples** ZOOM:VERTICAL:POSITION50 sets the vertical position to 50.  
 ZOOM:VERTICAL:POSITION? might return :ZOOM:VERTICAL:POSITION 23.90000, indicating that the vertical position is set to 23.9.

## ZOOM:VERTical:SCAle

This command sets or queries the zoom vertical scale factor around which the zoom waveform is displayed.

- Group** Zoom
- Syntax** ZOOM:VERTical:SCAle <NR3>  
 ZOOM:VERTical:SCAle?
- Arguments** <NR3> is the amount of expansion in the vertical direction in 1-2-5 increments.

**Examples** ZOOM:VERTICAL:SCALE 5 sets the vertical scale to 5.

ZOOM:VERTICAL:SCALE? might return :ZOOM:VERTICAL:SCALE 10, indicating that the vertical scale is set to 10.

## ZOOM:ZOOM<x>

This command resets the zoom transforms to default values for all traces of the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The ZOOM:REF<x>:VERTICAL:SCALE? query returns the current vertical and horizontal positioning and scaling of the display.

**Group** Zoom

**Syntax** ZOOM:ZOOM<x> {RESET|RESETLive}  
ZOOM:ZOOM<x>?

**Arguments** RESET resets the zoom transforms to default values for all traces of the specified zoom.

RESETLive resets the zoom transforms to default values for live traces of the specified zoom.

**Examples** ZOOM:ZOOM1 RESET resets the zoom transforms for all traces of Zoom1 to their default values

ZOOM:ZOOM1? might return :ZOOM:ZOOM1:STATE 1; SCROLLLOCK 1; CH1:HORIZONTAL POSITION 50.0000; SCALE 5; :ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000; SCALE 1.0000; :ZOOM:ZOOM1:CH2:HORIZONTAL:POSTION 50.0000; SCALE 5; :ZOOM:ZOOM1:CH2:VERTICAL:POSITION 0.000; SCALE 1.0000; :ZOOM:ZOOM1:CH3:HORIZONTAL:POSITION50.000; SCALE 5; :ZOOM:ZOOM1:CH3:VERTICAL... , indicating the Zoom1 transforms for all traces.

## ZOOM:ZOOM<x>:DCHAN:DISplay

This command sets or queries the display of the digital waveforms for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The digital waveforms affected vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, and D0 through D15. When zoom is set on any one of the waveforms from D0 through D15, it would apply to all of the waveforms (D0 through D15).

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:ZOOM<x>:DCHAN:DISPlay {<NR1> OFF ON} ZOOM:ZOOM<x>:DCHAN:DISPlay?
<b>Arguments</b>	<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.  OFF disables the specified zoom.  ON displays the specified zoom.
<b>Examples</b>	ZOOM:ZOOM1:DCHAN:DISPLAY ON displays the digital waveforms for the specified Zoom1.  ZOOM:ZOOM1:DCHAN:DISPLAY? might return :ZOOM:ZOOM1:DCHAN:DISPLAY 0, indicating that the display of the digital waveforms for the specified Zoom1 is disabled.

## ZOOM:ZOOM<x>:SCROLLLock

This command sets or queries Scroll Lock for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:ZOOM<x>:SCROLLLock {ON OFF <NR1>} ZOOM:ZOOM<x>:SCROLLLock?
<b>Arguments</b>	ON locks waveforms for the specified zoom window <x>.  OFF unlocks waveforms for the specified zoom window <x>.  <NR1> = 0 unlocks waveforms for the specified zoom window <x>; any other value locks waveforms for the specified zoom window <x>.
<b>Examples</b>	ZOOM:ZOOM1:SCROLLLOCK ON turns Zoom1 scroll lock on.  ZOOM:ZOOM1:SCROLLLOCK? returns either ON 1 or OFF 0 for Zoom1.

## ZOOM:ZOOM<x>:STATE

This command sets or queries the specified zoom on or off, where x is an integer from 1 to 4 representing the desired zoom window.

Default is Zoom1 on, Zoom2 - 4 off.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:ZOOM<x>:STATE {ON OFF <NR1>} ZOOM:ZOOM<x>:STATE?
<b>Arguments</b>	ON turns Zoom 1-4 on. OFF turns Zoom 1-4 off.  <NR1> = 0 disables the specified zoom; any other value enables the specified zoom.
<b>Examples</b>	ZOOM:ZOOM1:STATE ON turns Zoom1 on.  ZOOM:ZOOM2:STATE? might return :ZOOM:ZOOM2:STATE 1, indicating that Zoom2 is on.

## ZOOM:ZOOM<x>:<wfm>:DISplay

This command sets or queries the display of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The waveform affected is determine by <wfm> which can vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3, or Ref4.

<b>Group</b>	Zoom
<b>Syntax</b>	ZOOM:ZOOM<x>:<wfm>:DISplay {<NR1> OFF ON} ZOOM:ZOOM<x>:<wfm>:DISplay?
<b>Arguments</b>	<NR1> A zero disables the specified zoom; any other value enables the specified zoom.  OFF disables the specified zoom.  ON displays the specified zoom.

**Examples** ZOOM:ZOOM1:CH1:DISPLAY ON displays the specified zoom waveform.  
 ZOOM:ZOOM1:CH1:DISPLAY? might return :ZOOM:ZOOM1:CH1:DISPLAY 0, indicating the channel 1 zoom is disabled.

## ZOOM:ZOOM<x>:<wfm>:HORIZONTAL:POSITION

This command sets or queries the horizontal position of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The setting of the ZOOM:ZOOM<x>:SCROLLLock command determines the waveforms that are affected.

The waveform affected is determined by <wfm> which can vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3, or Ref4.

**Group** Zoom

**Syntax** ZOOM:ZOOM<x>:<wfm>:HORIZONTAL:POSITION <NR3>  
 ZOOM:ZOOM<x>:<wfm>: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 1× or greater.

**Examples** ZOOM:ZOOM1:CH1:HORIZONTAL:POSITION 50 sets the Zoom1 reference pointer at 50% of acquired waveform.  
 ZOOM:ZOOM1:CH1:HORIZONTAL:POSITION? might return :ZOOM1:CH1:HORIZONTAL:POSITION 50.0000, indicating that the Zoom1 reference pointer for Channel 1 is currently set at 50% of acquired waveform.

## ZOOM:ZOOM<x>:<wfm>:HORIZONTAL:SCALE

This command sets or queries the zoom horizontal scale factor of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

The waveform affected is determined by <wfm> which can vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3, or Ref4.

**Group** Zoom

<b>Syntax</b>	<code>ZOOM:ZOOM&lt;x&gt;:&lt;wfm&gt;:HORIZONTAL:SCALE &lt;NR3&gt;</code> <code>ZOOM:ZOOM&lt;x&gt;:&lt;wfm&gt;:HORIZONTAL:SCALE?</code>
<b>Arguments</b>	<NR3> is the amount of expansion in the horizontal direction in 1-2-5 increments.
<b>Examples</b>	<code>ZOOM:ZOOM1:CH1:HORIZONTAL:SCALE 5</code> sets the horizontal scale factor of Channel 1 to 5.  <code>ZOOM:ZOOM2:CH2:HORIZONTAL:SCALE?</code> might return: <code>ZOOM:ZOOM2:CH2:HORIZONTAL:SCALE 1</code> , indicating that Channel 2 has a horizontal scale factor of 1.

## ZOOM:ZOOM<x>:<wfm>:VERTICAL:POSITION

This command sets or queries the vertical position of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The setting of the `ZOOM:ZOOM<x>:SCROLLLOCK` command determines the waveforms that are affected. The <wfm> can be a channel, math, or reference waveform.

The waveform affected is determined by <wfm> which can vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3, or Ref4.

<b>Group</b>	Zoom
<b>Syntax</b>	<code>ZOOM:ZOOM&lt;x&gt;:&lt;wfm&gt;:VERTICAL:POSITION &lt;NR3&gt;</code> <code>ZOOM:ZOOM&lt;x&gt;:&lt;wfm&gt;:VERTICAL:POSITION?</code>
<b>Arguments</b>	<NR3> is the vertical position, expressed in divisions.
<b>Examples</b>	<code>ZOOM:ZOOM1:CH1:VERTICAL:POSITION 2</code> sets the Zoom1 vertical position to 2, which centers the zoom trace at the second division of the acquired trace.  <code>ZOOM:ZOOM1:CH1:VERTICAL:POSITION?</code> might return: <code>:ZOOM1:CH1:VERTICAL:POSITION 0.0000</code> , indicating that the Zoom1 trace for Channel 1 is centered at division 0 of the acquired trace.

## ZOOM:ZOOM<x>:<wfm>:VERTICAL:SCALE

This command sets or queries the zoom vertical scale of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

The waveform affected is determined by `<wfm>` which can vary from CH1, CH2, CH3, CH4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3, or Ref4.

**Group** Zoom

**Syntax** ZOOM:ZOOM<x>:<wfm>:VERTICAL:SCALE <NR3>  
ZOOM:ZOOM<x>:<wfm>:VERTICAL:SCALE?

**Arguments** <NR3> is the amount of vertical expansion or compression, which operates on a 1-2-5 sequence (for example, 1, 2, 5, 10, 20, 50, 100...). 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** ZOOM:ZOOM2:CH2:VERTICAL:SCALE 5 sets the vertical scale of Channel 2 to 5×.  
ZOOM:ZOOM2:CH2:VERTICAL:SCALE? might return  
:ZOOM2:CH2:VERTICAL:SCALE 2.0000, indicating that the vertical scale is 2×.



---

# Status and Events

The instrument provides a status and event reporting system for the GPIB 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

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.*

---

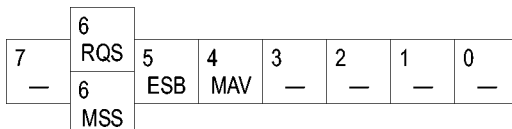
7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Table 3-1: SESR Bit Functions**

Bit	Function	
7 (MSB)	PON	Power On. Shows that the instrument was powered on. On completion, the diagnostic tests also set this bit.
6	URQ	User Request. Indicates that an application event has occurred. *See preceding 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.



**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 GPIB controller.
6	MSS	Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB	Event Status Bit. Shows that status is enabled and present in the SESR.
4	MAV	Message Available. Shows that output is available in the Output Queue.

**Table 3-2: SBR Bit Functions (cont.)**

Bit	Function
3	Not used.
2	Not used.
1-0	Not used.

**Enable**

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 as follows.

**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

**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

**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	—	—	—	—

## \*PSC Command

**\*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.** *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 more than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, “Queue Overflow.”

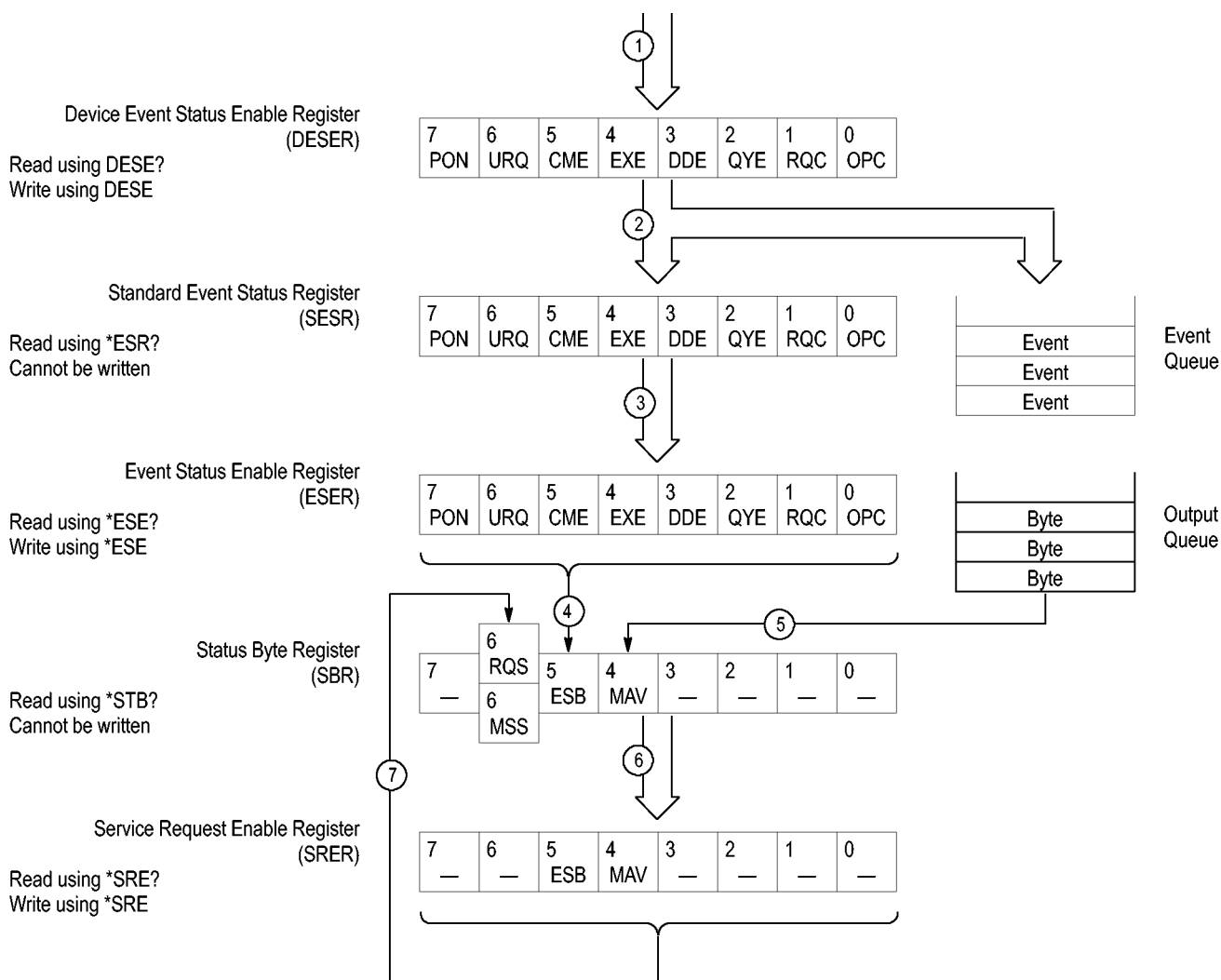
Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers with a description of the event). Reading an event removes it from 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 figure below shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.



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

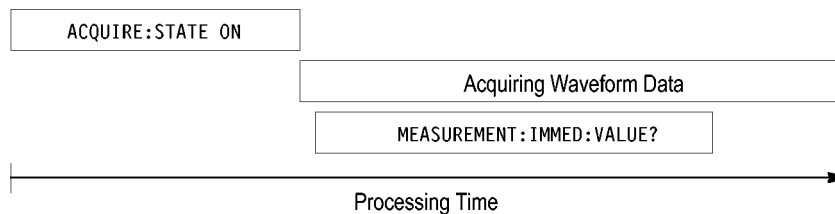
**Overview** Although most GPIB commands are completed almost immediately after being received by the instrument, some commands start a process that requires more time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may be a few seconds 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.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one gets processed. The instrument status and event reporting system provides ways to do this.

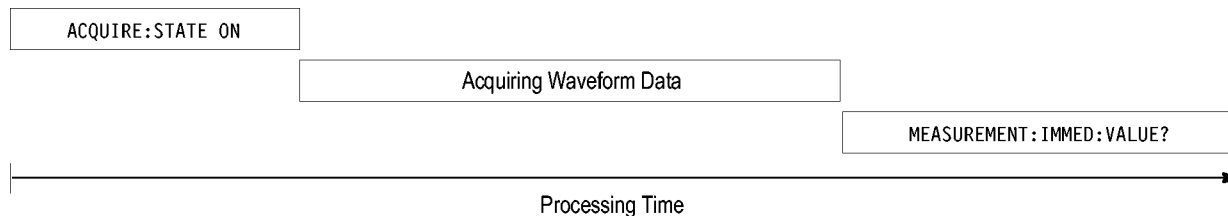
For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```
/** Set up conditional acquisition **/  
ACQUIRE:STATE OFF  
SELECT:CH1 ON  
HORIZONTAL:MODE: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  
/** Take amplitude measurement **/  
MEASUREMENT:IMMED:VALUE?
```

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.



To ensure that the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program.



You can use four commands to synchronize the operation of the instrument with your application program: \*WAI, BUSY, \*OPC, and \*OPC?

**\*WAI** 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
SELECT:CH1 ON
HORIZONTAL:MODE: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*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?

```

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 OPC operations in process are complete. If the input buffer becomes full, the

controller will be unable to write more commands to the buffer. This can cause a time-out.

**BUSY** 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
SELECT:CH1 ON
HORIZONTAL:MODE: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 more bus traffic.

#### \*OPC

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
SELECT:CH1 ON
HORIZONTAL:MODE: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 */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement.
*/
*OPC
While serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE

```

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, a Service Request will be generated.

The same command sequence using the \*OPC command for synchronization looks like this

```

/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON

```

HORIZONTAL:MODE:RECORDLENGTH 1000

ACQUIRE:MODE SAMPLE

ACQUIRE:STOPAFTER SEQUENCE

/\* Enable the status registers \*/

DESE 1

\*ESE 1

\*SRE 32

/\* 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

\*/

\*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:IMMED:VALUE

This technique is more efficient but requires more sophisticated programming.

\*OPC?

The \*OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. A time out could occur if you try to read the output queue before there is any data in it.

The same command sequence using the \*OPC? query for synchronization looks like this:

/\* Set up conditional acquisition \*/

ACQUIRE:STATE OFF

SELECT:CH1 ON

HORIZONTAL:MODE: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 CHI
/* Wait until the acquisition is complete before taking the measurement
*/
*OPC?
Wait for read from Output Queue.
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE

```

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.

## Messages

**Overview.** The information contained in the topic tabs 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 more 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 3-3: 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 3-4: Command Error Messages (CME Bit 5)**

<b>Code</b>	<b>Message</b>
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
111	Header separator error
112	Program mnemonic too long
113	Undefined header
114	Header suffix out of range
120	Numeric data error
121	Invalid character in numeric
123	Exponent too large
124	Too many digits
128	Numeric data not allowed
130	Suffix error
131	Invalid suffix
134	Suffix too long
138	Suffix not allowed
140	Character data error
141	Invalid character data
144	Character data too long
148	Character data not allowed
150	String data error
151	Invalid string data
158	String data not allowed
160	Block data error
161	Invalid block data
168	Block data not allowed
170	Command expression error
171	Invalid expression
178	Expression data not allowed
180	Macro error
181	Invalid outside macro definition

**Table 3-4: Command Error Messages (CME Bit 5) (cont.)**

<b>Code</b>	<b>Message</b>
183	Invalid inside macro definition
184	Macro parameter error
190	Communication option not present
191	Unable to open or parse arbitrary filter file

**Execution Error.** The following table lists the execution errors that are detected during execution of a command.

**Table 3-5: Execution Error Messages (EXE Bit 4)**

<b>Code</b>	<b>Message</b>
200	Execution error
201	Invalid while in local
202	Settings lost due to RTL
210	Trigger error
211	Trigger ignored
212	Arm ignored
213	Init ignored
214	Trigger deadlock
215	Arm deadlock
219	Trigger delay not available
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
225	Out of memory
226	Lists not same length
230	Data corrupt or stale
231	Data questionable
240	Hardware error
241	Hardware missing
244	Invalid printer selected
245	Hardcopy/Export operation timed out
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media

Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)

Code	Message
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
259	File system error
260	Execution expression error
261	Math error in expression
270	Execution macro error
271	Macro syntax error
272	Macro execution error
273	Illegal macro label
274	Execution macro parameter error
275	Macro definition too long
276	Macro recursion error
277	Macro redefinition not allowed
278	Macro header not found
280	Program error
281	Cannot create program
282	Illegal program name
283	Illegal variable name
284	Program currently running
285	Program syntax error
286	Program runtime error
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period found
2203	Measurement error, No period, second waveform
2204	Measurement error, Low signal amplitude
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, Waveform does not cross Mid Ref
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backward Mid Ref Crossing
2212	Measurement error, No negative crossing

**Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)**

<b>Code</b>	<b>Message</b>
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2218	Measurement error, Unused
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
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
2241	Waveform requested is invalid
2243	The measurement cannot be turned on because the source for the measurement is incompatible
2244	Source waveform is not active
2248	This reference waveform cannot be activated
2249	Reference deletion error, the reference waveform is in use for math
2250	Reference error, the reference waveform file is invalid
2251	Reference error, Waveform preamble does not match actual
2252	Reference error, Source waveform is not valid
2253	Reference error, Too many points received
2254	Reference error, Too few points received
2255	Reference error, destination reference is used by a math source
2256	Reference error, cannot reposition or rescale a reference waveform that was saved in FastAcq mode
2257	Reference warning, saving a previewed waveform results in an empty reference
2258	Reference warning, saving a FastAcq waveform may result in an empty reference
2259	File too big

**Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)**

<b>Code</b>	<b>Message</b>
2261	Calibration error, wait for the warm-up interval to expire before invoking SPC
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
2287	You have 4 free trials left for this application
2288	You have 3 free trials left for this application
2289	You have 2 free trials left for this application
2294	You have 1 free trial left for this application
2295	This is the last free trial for this application
2296	To guarantee accuracy the vertical amplifiers must use the same coupling and the same model probes
2297	CH1 and AUX cannot be selected as the A trigger event and the B trigger event respectively or vice versa
2298	CAL: failed to read NVram, will use uncalibrated factory defaults. The scope must be calibrated at the factory
2299	Failed to read SPC Calibration file, will use uncalibrated factory defaults until SPC has been performed
2309	Video Trigger does not allow selection of default trigger holdoff
2310	CAL: invalid command
2312	Deskew not applied when FastAcq is on
2313	Probe error, wait for the warmup interval to expire before doing probe calibration
2318	Consider system bandwidth when testing at this bit rate
2319	Deskew is re-enabled
2321	Histogram warning, histogram turned off because Zoom or FastFrame was enabled
2322	Histogram warning, turning histogram on turned other features off
2330	FastFrame time stamp query requested too many time stamps
2350	When FastFrame is active, the reference waveform source must be displayed
2351	Acquire Modes (Avg Env) incompatible with FastFrame
2352	Acquire Modes (Avg Env) incompatible with FastFrame. Set to Sample
2360	FastAcq warning, FastAcq turned Off
2370	Zoom warning, Zoom turned Off
2371	Zoom error, Zoom has circular definition



**Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)**

<b>Code</b>	<b>Message</b>
2372	Zoom error, Source zoom is not active
2400	Not enough memory available
2401	This channel cannot be activated
2402	Math/Meas/Histo have circular definition
2403	Cannot enter XYZ mode because Channel 4 is active (in use)
2404	XY waveform error, cannot activate channel while in XY mode
2405	XYZ waveform error, cannot activate channel while in XYZ mode
2410	Empty math string
2411	Syntax error in math string
2412	Semantic error in math string
2413	Math expression is too complex
2414	Spectral Function Syntax Error, can operate only on single Channel or Math source
2415	Spectral Function Semantic Error, cannot operate on static (Reference) sources
2420	Histogram cannot be performed on this type of waveform
2425	Mask counting cannot be performed on this type of wfm
2430	WfmDB cannot be built on this type of wfm
2470	Trigger error, cannot enter Random holdoff unless type is set to Edge
2471	Trigger error, cannot enter Random holdoff unless source is set to Channel 1-4
2472	Trigger error; cannot enter Random holdoff unless A->B Sequence is set to A Only
2473	Trigger T cannot be turned on when in XY display format mode
2474	Trigger slope not available, coerced to Rise
2475	Either trigger slope not available when in MultiScope AND mode
2476	Trigger type not available in MultiScope AND mode
2477	Pattern lock trigger not available when MultiScope triggering is enabled
2500	Setup error, file does not look like a setup file
2501	Setup warning, could not recall all values from external setup
2510	iCapture source selection failed; new source already in use
2511	Busses may not contain iCapture signals as members
2512	iCapture sources were reassigned in order to support serial bus triggering
2513	Clocked Parallel Bus clock source changed because previous source was added as member signal
2600	Mask testing complete
2601	Mask testing failed
2610	Standard Mask Autoset failed, not allowed for Math or Ref Waveforms

**Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)**

<b>Code</b>	<b>Message</b>
2611	Standard Mask Autaset failed, source of selected mask type not found
2612	Turn off Optical Filtering to allow this to work
2613	Optical Filtering not allowed
2614	Autaset is not allowed during Edit User Mask
2615	Mask setup error, file does not look like a setup file
2620	Mask error, too few points received
2700	Limit Test Error: Reference and source channel have different horizontal settings
2701	Limit Test Failure: Incompatible record lengths
2702	Template creation error: Source waveform not compatible
2703	Template creation error: Target reference is being used in an active limit test
2704	Limit Test Error: Reference is not a valid template
2705	Limit Test Failure: Cannot compare a reference to itself
2706	Limit Test Error: Disk Full
2707	Limit testing failed
2708	Select a source and template to compare before turning on the limit test
2740	No SMTP Server address has been entered
2741	Sending E-mail...
2742	Email on Trigger is disabled when Fast Acquisitions are enabled
2743	Email error; must have measurements active before selecting this option
2750	H-Bar cursors are not supported in xy mode
2751	V-Bar cursors are not supported in xy mode
2752	Cursor Error; the current source is invalid or inactive
2760	Mark limit reached
2761	No mark present
2762	Search copy failed
2763	Unable to recall the Marks
2764	Reference error, the reference waveform file is unsupported
2765	The Search cannot be turned on due to FastAcq or WfmDB mode
2766	Advanced Search Option not present
2767	Inputs CH1-CH4 incompatible; set state clock source to CH4
2768	Inputs incompatible; first set CH1-CH4 inputs to XXXX
2769	Search Mark sources must have matching record lengths
2770	Trigger Pulse source must be CH1-CH4 for this pulse class
2771	This Trigger Pulse class is not available for D0-D15
2772	D15-D0 is already used by A Event
2773	D15-D0 is already used by B Event

**Table 3-5: Execution Error Messages (EXE Bit 4) (cont.)**

Code	Message
2774	D15-D0 is already used by B Event Reset
2775	D15-D0 is already used by A Event qualification
2776	D15-D0 is already used by B Event qualification
2777	Busses cannot contain both live and reference signals
2778	Attempt to multiply add a signal to a bus

**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 3-6: Device Error Messages (DDE Bit 3)**

Code	Message
300	Device-specific error
310	System error
311	Memory error
312	PUD memory lost
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
330	Self test failed
350	Queue overflow (does not set DDE bit)

**System Event.** The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

**Table 3-7: 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
469	App button pressed

**Table 3-7: System Event Messages (cont.)**

Code	Message
470	TekScope minimized
471	TekScope restored
472	Application variable changed
485	Ch1 probe button pressed
486	Ch2 probe button pressed
487	Ch3 probe button pressed
488	Ch4 probe button pressed
489	Aux probe button pressed

**Execution Warning.** The following table lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

**Table 3-8: Execution Warning Messages (EXE Bit 4)**

Code	Message
532	Curve data too long, Curve truncated
533	Curve error, Preamble values are inconsistent
535	Curve error, Reference data sources not allowed in streaming or next-waveform mode
536	Curve error, Acquisition mode must be set to Run for streaming or next-waveform mode
537	Curve error, Cannot stream or next-waveform math waveforms while in roll mode
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
551	FASTAcq mode is active – deactivate to use math
590	Diagnostic warning, running GPIB diagnostic over GPIB may affect talk/listen program

**Internal Warning.** The following table shows internal errors that indicate an internal fault in the instrument.

**Table 3-9: Internal Warning Messages**

<b>Code</b>	<b>Message</b>
600	Internal warning
630	Internal warning, Input overload

**External Application Error.** The following table shows external application errors in the instrument.

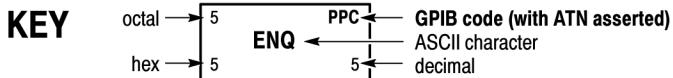
**Table 3-10: External Application Error**

<b>Code</b>	<b>Message</b>
2800	External application error



# Appendix A: Character Set

B7 B6 B5 BITS B4 B3 B2 B1	0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE			
0 0 0 0	0 NUL	20 DLE	40 SP	60 0	100 @	120 P	140 ' SA0	160 p SA16	0	0	0	0	0	0	0	0
0 0 0 1	1 SOH	21 DC1	41 !	61 1	101 A	121 Q	141 a SA1	161 q SA17	1	1	1	1	1	1	1	1
0 0 1 0	2 STX	22 DC2	42 "	62 2	102 B	122 R	142 b SA2	162 r SA18	2	2	2	2	2	2	2	2
0 0 1 1	3 ETX	23 DC3	43 #	63 3	103 C	123 S	143 c SA3	163 s SA19	3	3	3	3	3	3	3	3
0 1 0 0	4 EOT	24 DC4	44 \$	64 4	104 D	124 T	144 d SA4	164 t SA20	4	4	4	4	4	4	4	4
0 1 0 1	5 ENQ	25 NAK	45 %	65 5	105 E	125 U	145 e SA5	165 u SA21	5	5	5	5	5	5	5	5
0 1 1 0	6 ACK	26 SYN	46 &	66 6	106 F	126 V	146 f SA6	166 v SA22	6	6	6	6	6	6	6	6
0 1 1 1	7 BEL	27 ETB	47 ' LA7	67 7	107 G	127 W	147 g SA7	167 w SA23	7	7	7	7	7	7	7	7
1 0 0 0	8 BS	30 CAN	50 ( LA8	70 8	110 H	130 X	150 h SA8	170 x SA24	8	8	8	8	8	8	8	8
1 0 0 1	9 HT	31 EM	51 ) LA9	71 9	111 I	131 Y	151 i SA9	171 y SA25	9	9	9	9	9	9	9	9
1 0 1 0	A LF	32 SUB	52 * LA10	72 :	112 J	132 Z	152 j SA10	172 z SA26	A	A	A	A	A	A	A	A
1 0 1 1	B VT	33 ESC	53 + LA11	73 ; LA27	113 K	133 [	153 k SA11	173 { SA27	B	B	B	B	B	B	B	B
1 1 0 0	C FF	34 FS	54 , LA12	74 < LA28	114 L	134 \	154 l SA12	174   SA28	C	C	C	C	C	C	C	C
1 1 0 1	D CR	35 GS	55 - LA13	75 = LA29	115 M	135 ]	155 m SA13	175 } SA29	D	D	D	D	D	D	D	D
1 1 1 0	E SO	36 RS	56 . LA14	76 > LA30	116 N	136 ^	156 n SA14	176 ~ SA30	E	E	E	E	E	E	E	E
1 1 1 1	F SI	37 US	57 / LA15	77 ? UNL	117 O	137 -	157 o SA15	177 RUBOUT (DEL)	F	F	F	F	F	F	F	F
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES	TALK ADDRESSES	SECONDARY ADDRESSES OR COMMANDS											



**Tektronix**  
 REF: ANSI STD X3.4-1977  
 IEEE STD 488.1-1987  
 ISO STD 646-2973





# Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

*CAL	ALL	ATTEN350X	AUTOBright
*CLS	ALLEV	ATTEN35X	AUTOReflEvel
*DDT	ALLFields	ATTEN3X	AUTOSet
*ESE	ALLLines	ATTEN3_5X	AUTOZero
*ESR	ALLMatched	ATTEN400X	AUX
*IDN	ALLOcate	ATTEN40X	AUXIn
*LRN	ALLQString	ATTEN450X	AUXLeve1
*OPC	AMI	ATTEN45X	AUXiliary
*OPT	AMPlitude	ATTEN4X	AUXout
*PSC	AND	ATTEN4_5X	AVerage
*PUD	ANNOtation	ATTEN500X	B
*RCL	AOFFSet	ATTEN50X	B11t;x>
*RST	APPLYtoall	ATTEN550X	B3ZS
*SAV	AREA	ATTEN55X	B6ZS
*SDS	ASCIi	ATTEN5X	B8ZS
*SRE	ATAG1	ATTEN5_5X	BACKwards
*STB	ATAG2	ATTEN600X	BADCHARS
*TRG	ATAG3	ATTEN60X	BANDwidth
*TST	ATARXG1	ATTEN650X	BASE
*WAI	ATARXG2	ATTEN65X	BAUD
0	ATARXG3	ATTEN6X	BEEP
1	ATATXG1	ATTEN6_5X	BEL1
100	ATATXG2	ATTEN700X	BETween
10Bit	ATATXG3	ATTEN70X	BILeve1custom
50	ATRIGger	ATTEN750X	BINary
7Bit	ATTEN1000X	ATTEN75X	BIT
80	ATTEN100X	ATTEN7X	BITCOUNT
A	ATTEN10X	ATTEN7_5X	BITDURATION
ABSolute	ATTEN125X	ATTEN800X	BITOrder
AC	ATTEN12_5X	ATTEN80X	BITRATE
ACcept	ATTEN150X	ATTEN850X	BIT_Nr
ACKMISS	ATTEN15X	ATTEN85X	BLACKANDWhite
ACQDURATION	ATTEN175X	ATTEN8X	BLACKMANHarris
ACQLENGTH	ATTEN17_5X	ATTEN8_5X	BMP
ACquire	ATTEN1X	ATTEN900X	BN_Fmt
ACTIVE	ATTEN1_25X	ATTEN90X	BOFFSet
ACTUa1	ATTEN1_5X	ATTEN950X	BOTH
ADDR10	ATTEN1_75X	ATTEN95X	BOX
ADDR7	ATTEN200X	ATTEN9X	BOXPcnt
ADDRANDDATA	ATTEN20X	ATTEN9_5X	BTRIGger
ADDRess	ATTEN250X	ATTempts	BURst
ADVANCEafter	ATTEN25X	AUTHLogin	BUS
ALIGN	ATTEN2X	AUTHPassword	BUS1
ALIGNCHARacter	ATTEN2_5X	AUTO	BUS10
ALIGNPRIMitive	ATTEN300X	AUTOAdjust	BUS11
ALias	ATTEN30X	AUTOBitrate	BUS12

BUS13	COLUMN	DATASource	DS0Timing
BUS14	COMMunication	DATE	DS1
BUS15	COMPLetion	DAYS	DS1A
BUS16	COMPLeted	DB	DS1C
BUS2	COMbination	DBM	DS2
BUS3	COMMonmode	DC	DS2RATECoax
BUS4	COMpare	DCHAN	DS2RATESymmetrical
BUS5	CONDition	DCREJect	DS3
BUS6	CONNEction	DDR	DS4NA
BUS7	CONSTant	DDR2	DS4NA_Max
BUS8	CONSUMER	DDR3	DURATION
BUS9	CONTROL	DDRMemory	D<lt;x>
BUSY	COPY	DDRREADwrite	Dx
BY	COUNT	DDRRead	E1
BYPass	COUPLing	DDRwrite	E1Coax
BYT_Nr	CQ	DECode	E1Symmetrical
BYT_Or	CQ0	DECimal	E2
Block	CQ<lt;x>	DEFAULT	E3
B<lt;x>	CREATE	DEFine	E4
Bx	CRMs	DEGAUSS	E4_0
CALProbe	CROSSHair	DEGrees	E4_1
CALibrate	CStActive	DELEte	ECL
CAN	CSLevel	DELIMiter	EDGE
CANH	CSMode	DELTATime	EDGE2
CANL	CSSource	DELTA	EDGE[1]
CAPS	CURSOr	DELAy	EDGE<lt;x>
CARea	CURVENext	DELAyed	EIGHTytwenty
CARet	CURVESTream	DESE	EITHER
CATaLog	CURVE	DESKEW	EMBOSS
CENTER	CUSTOM	DESTination	EMail
CENTERed	CWD	DETAiled	ENABLE
CH1	D0	DIAG	ENCdg
CH14	D1	DIFFerential	END
CH2	D10	DIGITALALLQString	ENDevent
CH3	D11	DIGTraces	ENET100
CH4	D12	DIGital	ENET100BCX_ATP2
CHARACTER	D13	DIR	ENET100BCX_ATP3
CHECKSUMDR	D14	DIREction	ENET100BCX_NTP2
CH<lt;x>	D15	DISPARITY	ENET100FX
CHx	D15CH1	DISPCOUNT	ENET100STP
CJTP	D2	DISPRATE	ENET100UTP
CLASS	D3	DISTDuty	ENET1250
CLEAR	D3D4	DISable	ENETXAUI
CLOCKCoax	D4	DISPlay	ENETXAUI2
CLOCKSymmetrical	D5	DISPlaymode	ENETXAUI_Far
CLOCK	D6	DIVisions	ENETXAUI_Near
CMDBatch	D7	DMOFFSet	ENGRave
CMEan	D8	DONE	ENHanced
CMI	D9	DOTS	ENTERSWindow
CMOFFSet	DALL	DREF	ENV
CODE	DASHed	DS0Contra	ENVELOpe
COEFFS	DATA	DS0Double	EOF
COLOR	DATABits	DS0Single	ERROR

ERRORLIMIT	FC2125E_ADT	FREETRIALBurn	HD1080P50
ERRORdetector	FC2125E_AGR	FREETRIALsleft	HD1080P60
ET	FC2125E_AGT	FREQuency	HD1080SF24
EVEN	FC2125E_NBT	FREV	HD480P60
EVENT	FC2125E_NDT	FROMSearchx	HD576P50
EVENTS	FC2125E_NGT	FRom	HD720P30
EVMsg	FC266	FST	HD720P50
EVQty	FC266E	FST1	HD720P60
EXECUTE	FC4250	FST2	HD875I60
EXITSwindow	FC4250E_ABR	FST3	HDB3
EXponent	FC4250E_ABT	FST4	HDBlt;x>
EXPort	FC4250E_ADR	FST5	HDELTA
EXTatten	FC4250E_ADT	FST6	HDR
EXTDBatten	FC4250E_AGR	FULLNomenue	HEADER
EXTENDED	FC4250E_AGT	FULLSCREEN	HEIght
EXTINCTDB	FC4250E_NBT	FULl	HERtz
EXTINCTPCT	FC4250E_NDT	FUNCTION	HEX
EXTINCTRATIO	FC4250E_NGT	FW1394BS1600B	HEXadecimal
EXTUnits	FC531	FW1394BS1600BT1	HFRej
EXternal	FC531E	FW1394BS1600BT2	HFTP
EYE	FFWD	FW1394BS400B	HIACQRATE
EYEHeight	FIELD	FW1394BS400BT1	HIGH
EYEWidth	FIELD1	FW1394BS400BT2	HIGHLIGHTHits
EYEdiagram	FIELD2	FW1394BS800B	HIGHLOWX
FACTory	FIFtyfifty	FW1394BS800BT1	HIGHLimit
FAILURES	FILE	FW1394BS800BT2	HIRes
FAILure	FILEFormat	FWD	HISTogram
FALL	FILEName	FuLl	HITS
FALSE	FILESystem	G703DS1	HLS
FALLing	FILTER	G703DS3	HOLDTime
FAST	FILEpath	GAIN	HOLDoff
FASTAcq	FIRSt	GATEPOS	HORizontal
FASTERthan	FIVE	GATEWIDTH	HORizontal[
FASTframe	FLATTOP2	GATing	HOSTwanted
FAStest	FOCUS	GAUSSian	HOURS
FC1063	FONTCOLOR	GDDR3	HPOS
FC1063Draft	FONTEffect	Generalcall	HPOSlt;x>
FC1063E	FONTName	GLitch	HSCALE
FC1063E_ABR	FONTSize	GND	HST
FC1063E_ABT	FONTStyle	GRaticule	HST1
FC1063E_ADR	FORCEDRange	GRId	HST2
FC1063E_ADT	FORCe	GROUPDelay	HST3
FC1063E_AGR	FORMat	HALT	HST4
FC1063E_AGT	FORwards	HAMming	HST5
FC1063E_NBT	FP	HANNing	HST6
FC1063E_NDT	FPBinary	HARDCopy	HTRIGPOS
FC1063E_NGT	FRAME	HBars	HWAccountant
FC133	FRAMELock	HBWstraightflex	HYSterisis
FC133E	FRAMEStart	HD1080I50	I2C
FC2125	FRAMESTOP	HD1080I60	ICAPture
FC2125E_ABR	FRAMEStart	HD1080P24	ID
FC2125E_ABT	FRAMetype	HD1080P25	IDANDDATA
FC2125E_ADR	FREE	HD1080P30	IDENTifier

IMAGEView	LINE_X	MEAS1	NOvershoot
IMAGE	LIST	MEAS2	NR1
IMMEDState	LIVE	MEAS3	NR1QString
IMMed	LOCK	MEAS4	NR2
INdependent	LOCKLen	MEAS5	NR3
INF2_5G	LOCKOffset	MEAS6	NRF
INF2_5GE	LOG	MEAS7	NRZ
INFINIBAND	LOGic	MEAS8	NR_FR
INFINite	LONG	MEASUREMENT	NR_Pt
INFPersist	LOOP	MEAS<lt;x>	NTSC
INHERIT	LOOPS	MEDian	NUMACq
INIT	LOW	MEDium	NUMAVg
INITIALCRCVAlue	LOWLimit	METHod	NUMCOEFFS
INKSaver	LOWERthreshold	MFTP	NUMEMails
INPUT	LPDDR	MHZ10	NUMERIC
INPUTMode	LSB	MHZ100	NUMEnv
INSIDEGreater	LSSTstatus<lt;x>	MID	NUMFRAMESAcquired
INSide	LVLsrcpreference	MID<lt;x>	NUMFRAMES
INTENSIFied	MAG	MINCas	NUMITEMS
INTENSITY	MAGnivu	MINimum	NUMSAmPles
INTERLaced	MAIn	MINMax	NUMX
INTERNAL	MAIn]	MINUSOne	NUMY
INTERPEightbit	MANChester	MINUTES	NWIdth
INTERPRatio	MANUAL	MISO	No
INVERTed	MARK	MISOMOSI	OC1
INVert	MARKSINCOLumn	MKDir	OC12
IRE	MARgin	MLT3	OC3
ISCLOCKED	MASK	MODE	OC48
IT	MASKhighlight	MONOGRAY	OC48_FEC
ITEM	MASKPRE	MONOGREEN	OCCurs
JPEG	MAT1	MORELimit	ODD
KAISERBessel	MATH1	MOREThan	OFF
LABEL	MATH14	MOSI	OFFGRaticule
LABell<lt;x>	MATH2	MRTTime	OFFSETAdj
LANDscape	MATH3	MSB	OFFSet
LARge	MATH4	MULTiscope	ON
LAST	MATHArbf<lt;lt;x>	MULTipleframes	ONEfifty
LAYout	MATHCOLOR	MV	ONGRaticule
LBPT	MATHCad	NAME	ONLYOVERlay
LEARN	MATHVAR	NAND	OPTIMnoise
LENGTH	MATH<lt;x>	NCross	OPTION
LESSLimit	MATHx	NDuty	OR
LESSThan	MATHy	NEGative	ORR
LEVEL	MATLab	NEWpass	OTHER
LFRej	MAXALIGNs	NEXT	OUT
LFTP	MAXCas	NO	OUTSIDEGreater
LIMITrecordlen	MAXFRAMES	NOCARE	OUTSide
LIMit	MAXSAMPLERate	NOISERej	OVERLOAD
LINE	MAXSize	NOISe	OVERlay
LINEAr	MAX_AP	NONE	OWNER
LINEPeriod	MAXimum	NOR	PAL
LINES	MEAN	NORMALIZED	PALETTE
LINEStyle	MEANSTDdev	NORMAL	PARAllel

PARity	PROGressive	REFLEVELoffset	ROLE
PASS	PRObestate	REFLevel	ROSC
PASSword	PTOP	REFout	RP
PATTERNBITS	PT_Fmt	REF<lt;x>	RPBinary
PATTERNNAME	PT_Order	REFx	RS232
PATtern	PT_Off	REJect	RS232c
PBASE	PULSEForm	RELoad	RT
PCIEXPRESS	PULse	REM	RUN
PCIEXPRESS_Rcv	PWIdth	REMOte	RUNSTop
PCIEXPRESS_Xmit	QFActor	REName	RUNT
PCIExpress	QString	REPEATstart	RWINADDR
PCIExpress2	QUALification	REPeat	RWINCLude
PCROSS	QUALify	RESBW	RX
PCTCROSS	QUERYRESPonse	RESET	S8B10B
PCX	RADIUS<lt;x>	RESETLive	S8BPATtern
PDuty	RADians	RESULT	SAMPLERate
PEAKHits	RANDOM	RESULTS	SAMPlingmode
PEAKdetect	RANge	RESistance	SAMPle
PENDING	RATDELta	REV	SAS1_5
PERCent	RATE	RI	SAS1_5_CR
PERIod	RATE1	RIBinary	SAS1_5_CR_AASJ
PERSistence	RATE2	RIO_1G	SAS1_5_IR
PHASE	RATE3	RIO_1_5G	SAS1_5_IR_AASJ
PK2Pk	RATE32Mbit	RIO_2G	SAS1_5_SATA
PKPKJitter	RATE4	RIO_500M	SAS1_5_XR
PKPKNoise	RATE5	RIO_750M	SAS1_5_XR_AASJ
PLOCK	RATE97Mbit	RIO_DRV1G	SAS3_
PLOCK BITRate	RATIO<lt;x>	RIO_DRV1_5G	SAS3_0_CR
PLUSone	RATIO	RIO_DRV2G	SAS3_0_CR_AASJ
PNG	RDELta	RIO_DRV500M	SAS3_0_IR
POINTS	RDMINUS	RIO_DRV750M	SAS3_0_IR_AASJ
POLARCoord	RDPLUS	RIO_EDRV1G	SAS3_0_SATA
POLarity	READ	RIO_EDRV1_5G	SAS3_0_XR
PORT	READFile	RIO_EDRV2G	SAS3_0_XR_AASJ
PORTrait	READOUT	RIO_EDRV500M	SAS6_0
POSITION<lt;x>	READOuts	RIO_EDRV750M	SATA1_5
POSITIVE	READY	RIO_RCV1G	SATA3_0
Position	RECALL	RIO_RCV1_5G	SATA6_0
Position<lt;x>	RECORDview	RIO_RCV2G	SAVE
POS<lt;x>	RECOVered	RIO_RCV500M	SAVEWFM
POVershoot	RECOVlength	RIO_RCV750M	SCAN
PRESAMPBITS	RECTANGular	RIO_SERIAL_1G	SCALE
PREvious	RECTX<lt;x>	RIO_SERIAL_2G	SCLK
PRINTER	RECTY<lt;x>	RIO_SERIAL_2_5G	SCRAMBLED
PRInt	RECTangular	RIO_SERIAL_3G	SCREEN
PROBE	REF	RISe	SCREENSAVER
PROBEContro1	REF1	RISetime	SCREENSAVERDELAY
PROBECa1	REF14	RISing	SCREEntext
PROBEFunc	REF2	RMDir	SCROLL
PRODELta	REF3	RMS	SCROLLLock
PRODUCT<lt;x>	REF4	RMSJitter	SDASHed
PRODUCer	REFCOLOR	RMSNoise	SEARCH
PRODUCT	REFC1k	ROLL	SEARCH<lt;x>

SEARCHtotrigger	SLOWERthan	STRing	TRIGgeron
SECAM	SLOpe	STS1	TRIGgertosearch
SECONDS	SMAlL	STS1Eye	TRILevelcustom
SEGl<t;m>	SMTPPort	STS1Pulse	TRIGger
SELECTED	SMTPServer	STS3	TRUE
SElect	SNAP	STS3_Max	TTL
SENSITivity	SNRatio	STYle	TWEnty
SEQUENTIAL	SOF	SUBITEMS	TWOfifty
SEquence	SOLID	SUBSYS	TX
SERIAL	SOURCE	SUMFrame	TYPE
SERIALTRIG	SOURCES	SUPPress	UNDEFINED
SERnumber	SOURcel<t;x>	SYMBOL	UNDO
SET	SPAN	SYMBOLic	UNITString
SETCHANGE	SPC	SYNC	UNIts
SETHold	SPECTraL	SYNCIntervaL	UNLOCK
SETLOGIC	SPEED	SYNCPATtern	UNWRap
SETLevel	SPI	SYNCSOURces	UPPerthreshold
SETRole	SPLit	TEKEXponential	USER
SETSTATE	SPREADSHEETCv	TEKLink	USERMask
SETTime	SPREADSHEETxt	TEKSecure	VALUEmean
SETUp	SRCDependent	TEMPERature	VALue
SFI5_2	SRCIndependent	TEMPlate	VARl<t;x>
SFI5_3	SRibinary	TENbit	VARpersist
SFI5_RCVBCLK2	SRPbinary	TERmination	VBArs
SFI5_RCVBCLK3	SRQ	TEST	VDELTA
SFI5_RCVBDATA2	SS	TESTS	VECTors
SFI5_RCVBDATA3	SSC	TFI5_2	VERBoSe
SFI5_RCVDCLK2	STABle	TFI5_3	VERT_INDEX
SFI5_RCVDCLK3	STANDARD	THDELta	VERTical
SFI5_RCVDDATA2	START	THETA<t;x>	VFields
SFI5_RCVDDATA3	STARTevent	THRESHold	VIDEO270
SFI5_XMITACLK2	STARTbyte	TIFF	VIDEO292M
SFI5_XMITACLK3	STATE	TIME	VIDEO360
SFI5_XMITADATA2	STATIstics	TIMEOut	VIDEo
SFI5_XMITADATA3	STATUS	TIMEStamp	VIEW
SFI5_XMITCCLK2	STAYSHigh	TIPType	VIOLATIONS
SFI5_XMITCCLK3	STAYSLow	TO	VLines
SFI5_XMITCDATA2	STDdev	TOGGLE	VOFFSet
SFI5_XMITCDATA3	STM0_0	TOLerance	VPOS
SFPbinary	STM0_1	TOTAL	VSCAle
SHADow	STM0_CMI	TRACK	VSROC192
SHIFted	STM0_HDBX	TRACKing	WAIT
SHORT	STM1E	TRANSITION	WAVEFORMS
SHOWREmote	STM1E_0	TRIG	WAVEform
SIGMA1	STM1E_1	TRIGBar	WAVFRMStream
SIGMA2	STOP	TRIGIF	WAVFrm
SIGMA3	STOPAfter	TRIGT	WEIghting
SIGNAL	STOPOnviation	TRIGTOSAMP	WFID
SIGType	STOPWHEN	TRIGger	WFMDB
SINX	STORE	TRIGgerAPULse	WFMInpre
SIXSigmajit	STRESS	RUNTLOGicINPUT	WFMOutpre
SIZE	STROBE	TRIGgerAPULse	WFMPre
SKIPSETPRIMitive	STROBESource	WINDowLOGicINPUT	WHEN

WIDERthan	Y	decodeFileName	makeEnumListbus
WIDTH	YDELta	directorypath	_rs232Source
WINDow	YES	dynamicRangenR3	makeEnumListtrigMgrB
WIThin	YMUlt	fileName	_setupholdTriggerLogic
WRITE	YOff	filepath	makeEnumListtrigMgr
WRITEFile	YPOS	filepathGPIb	_busRS232TriggerOn
X	YPOSITION<lt;x>	filepathREFx	newworking
XDELta	YT	filepathdata	directorypath
XINcr	YUNit	<lt;wfm>	nr1
XPOS	YZero	makeEnumListbus	oldfilepath
XPOSITION<lt;x>	Yes	_rs232DisplayMode	newfilepath
XSMAl1	Y<lt;x>	makeEnumListbus	rangeName
XUNit	ZOOM<lt;x>	_rs232PacketDelimiter	sourcefilepath
XY	ZOOMx	makeEnumListbus	destinationfilepath
XYZ	ZOOM	_rs232Parity	wfm
XZerO	asccurve	makeEnumListbus	wfmQString
X<lt;x>	busName	_rs232Polarity	wfmREFx





# Appendix C: Factory Default Setup Values

## Default Setup

The following tables list the default setup values by command group. The instrument sets these values when you press the front panel DEFAULT SETUP button or send the FACTORY command. Only those commands that have values set by the DEFAULT SETUP function are listed in these tables.

---

**NOTE.** The following commands do not affect the DATA parameters: TEKSecure, \*RST, RECALL:SETUP FACTORY. These are the initial factory default settings for DATA :DATA:DESTINATION REF1;ENCDG RIBINARY;SOURCE CHI;START 1;STOP 5000;FRAMESTART 1;FRAMESTOP 5000.

---

---

**NOTE.** Find the most up-to-date values for your instrument and software by pressing default setup and looking at the instrument to get the default values.

---

### Acquisition Default Values

The following table lists the Acquisition factory default setup values.

Command	Default Values
ACQUIRE:MODE	SAMPLE
ACQUIRE:NUMAVG	16
ACQUIRE:NUMENV	10
ACQUIRE:NUMSAMPLES	16000
ACQUIRE:SAMPLINGMODE	IT
ACQUIRE:STATE	1 (ON)
ACQUIRE:STOPAFTER	RUNSTOP
FASTACQ:STATE	0

### Alias Default Values

The following table lists the Alias factory default setup values.

Command	Default Values
ALIAS:STATE	0 (OFF)

### Bus Default Values

The following table lists the Bus factory default setup values.

Command	Default Values
BUS:B<x>:DISPLAY:DECODE:STATE	OFF
BUS:B<x>:DISPLAY:TYPE	Busform
BUS:B<x>:I2C:CLOCK:SOURCE	D1

Command	Default Values
BUS:B<x>:I2C:DATA:SOURce	D0
BUS:B<x>:I2C:RWINADDR	No
BUS:B<x>:LABel	"Bus 1"
BUS:B<x>:PARAllel:CLOCK:EDGE	Rising
BUS:B<x>:PARAllel:CLOCK:SOURce	D15
BUS:B<x>:PARAllel:ISCLOCKED	No
BUS:B<x>:POSition	0.0div
BUS:B<x>:SPI:BITOrder	MSB
BUS:B<x>:SPI:CLOCK:POLarity	Rising
BUS:B<x>:SPI:CLOCK:SOURce	D2
BUS:B<x>:SPI:DATA:POLarity	Normal
BUS:B<x>:SPI:DATA:SIZE	8
BUS:B<x>:SPI:DATA:SOURce	D0
BUS:B<x>:SPI:SElect:POLarity	Active low
BUS:B<x>:SPI:SElect:SOURce	D1
BUS:B<x>:TYPE	Parallel
SElect:B<x>	OFF

**Cursor Default Values**

The following table lists the Cursor factory default setup values.

Command	Default Values
CURSor:FUNcTion	VBARS
CURSor:HBArs:POSITION<x>	1 = 300.0000E-3 2 = -300.0000E-3
CURSor:LINEStYle	SDASHED
CURSor:MODE	INDEPENDENT
CURSor:SCREEN:StYle	LINES
CURSor:SOURce<x>	CH1
CURSor:STATE	0
CURSor:VBArS:POSITION<x>	<b>For instruments &gt;= 12.5 GHz</b> 1 = -8.0000E-9 2 = 8.0000E-9 <b>For instruments &gt; 3.5 GHz and &lt; 12.5 GHz</b> 1 = -16.0000E-9 2 = 16.0000E-9 <b>For instruments &lt;= 3.5 GHz</b> 1 = -40.0000E-9 2 = 40.0000E-9

Command	Default Values
CURSor:VBArS:UNIts	SECONDS
CURSor:WAVEform:POSition<x>	<b>For instruments &gt;= 12.5 GHz</b> 1 = -8.0000E-9 2 = 8.0000E-9 <b>For instruments &gt; 3.5 GHz and &lt; 12.5 GHz</b> 1 = -16.0000E-9 2 = 16.0000E-9 <b>For instruments &lt;= 3.5 GHz</b> 1 = -40.0000E-9 2 = 40.0000E-9
CURSor:WAVEform:STYle	LINE_X
CURSor:WAVEform:UNIts	BASE
CURSor:XY:READOUT	RECTANGULAR
CURSor:XY:RECTX<x>	0.0000
CURSor:XY:RECTY<x>	0.0000

### Diagnostics Default Values

The following table lists the Diagnostics factory default setup values.

Command	Default Values
DIAG:LEVEL	SUBSYS

<sup>1</sup> The factory default is not restored by Default Setup, TekSecure, or \*RST.

### Digital Default Values

The following table lists the Digital factory default setup values.

Command	Default Values
DISplay:DIGital:HEIght	Large
D<x>:POSition	-4.66div
D<x>:THRESHold	1.4V

### Display Control Default Values

The following table lists the Display factory default setup values.

Command	Default Values
DISplay:CLOCK	1
DISplay:COLOr:MATHCOLOr	DEFAULT
DISplay:COLOr:PALEtte:IMAGEView 1	TEMPERATURE
DISplay:COLOr:PALEtte:RECORDView	NORMAL
DISplay:COLOr:PALEtte:USER:CARet	150,50,100

Command	Default Values
DISplay:COLOr:PALEtte:USEr:CH<x> 1	1 = 180,50,100 2 = 300,50,100 3 = 60,50,100 4 = 240,50,100
DISplay:COLOr:PALEtte:USEr:GRAticule 1	On DPO7000 series of products it is 320,10,0 On MSO70000 series of products it is 165,50,15
DISplay:COLOr:PALEtte:USEr:HIStogram 1	On DPO7000 series of products it is 320,10,0 On MSO70000 series of products it is 320,50,100
DISplay:COLOr:PALEtte:USEr:MASK 1	On DPO7000 series of products it is 320,10,0 On MSO70000 series of products it is 0,40,100
DISplay:COLOr:PALEtte:USEr:MASKHighlight 1	On DPO7000 series of products it is 320,10,0 On MSO70000 series of products it is 140,50,100
DISplay:COLOr:PALEtte:USEr:MATH<x>	1 = 160,50,100 2 = 40,60,100 3 = 120,60,100 4 = 195,50,100
DISplay:COLOr:PALEtte:USEr:REF<x> 1	1 = 0,90,0 2 = 0,90,100 3 = 60,90,100 4 = 240,90,100
DISplay:COLOr:REFCOLOr	DEFAULT
DISplay:FILTer	SINX
DISplay:FORMat	YT
DISplay:GRAticule	FULL
DISplay:INTENSITy:WAVEform:IMAGEView	75.0000
DISplay:INTENSITy:WAVEform:RECORDView	75.0000
DISplay:PERStence	OFF
DISplay:SCREExt:LABel<x>:NAME	""
DISplay:SCREExt:LABel<x>:XPOS	100

Command	Default Values
DISplay:SCREExt:LABel<x>:YPOS	1 = 5 2 = 20 3 = 35 4 = 50 5 = 65 6 = 80 7 = 95 8 = 110
DISplay:SCREExt:STATE	0
DISplay:STYle	VECTORS
DISplay:TRIGBar	SHORT
DISplay:TRIGT	1
DISplay:VARpersist	500.0000E-3

<sup>1</sup> 1. the factory default is not restored by Default Setup, TekSecure, Factory, or \*RST. The factory default is restored by the DISplay:COLOR:PALETTE:USER Reset command.

## E-mail Default Values

The following table lists the E-mail setup values.

Command	Default Values
EMail:ATTempts <sup>1</sup>	1
EMail:AUTHLogin	""
EMail:FRom <sup>1</sup>	"DPO7104_<Instrument Serial Number>"
EMail:HOSTwanted <sup>1</sup>	""
EMail:IMAGe	0
EMail:MASK	0
EMail:MAXSize	1
EMail:MEASUrement	0
EMail:NUMEMails	1
EMail:SMTPPort <sup>1</sup>	25
EMail:SMTPServer	""
EMail:TIMEOut <sup>1</sup>	30
EMail:TO	""
EMail:TRIGger	0
EMail:WAVEform	0

<sup>1</sup> These factory defaults are not reset to factory defaults by Default Setup.

**Error Detector Default Values**

The following table lists the Error Detector setup values.

<b>Command</b>	<b>Default Values</b>
TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDMINUS	"0011111010"
TRIGger:A:SERIAL:ERRORdetector:ALIGNCHARacter:RDPLUS	"1100000101"
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDMINUS:PATtern	SATA default is "0011111010010101010101010101010010011100"
TRIGger:A:SERIAL:ERRORdetector:ALIGNPRIMitive:RDPLUS:PATtern	SATA default is "110000010101010101010101010101011101100011"
TRIGger:A:SERIAL:ERRORdetector:BIT:SYNCPATtern	K28.5, "0011111010", and "1100000101"
TRIGger:A:SERIAL:ERRORdetector:DURATION:COUNT	-1
TRIGger:A:SERIAL:ERRORdetector:ENable	OFF
TRIGger:A:SERIAL:ERRORdetector:FRAME:INITIALCRCVALue	SATA
TRIGger:A:SERIAL:ERRORdetector:MAXALIGNs	8
TRIGger:A:SERIAL:ERRORdetector:SCRAMBLED	ON
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDMINUS:PATtern	"0011111010001111101000111110101010101010"
TRIGger:A:SERIAL:ERRORdetector:SKIPSETPRIMitive:RDPLUS:PATtern	"1100000101110000010111000001011010101010"
TRIGger:A:SERIAL:ERRORdetector:SSC	ON
TRIGger:A:SERIAL:ERRORdetector:STANDARD	SATA
TRIGger:A:SERIAL:ERRORdetector:STOPWHEN	MANUAL
TRIGger:A:SERIAL:ERRORdetector:TYPE	BIT

**Hard Copy Default Values**

The following table lists the Hardcopy factory default setup values.

<b>Command</b>	<b>Default Values</b>
HARDCopy:FILENAME	""
HARDCopy:PORT	FILE

**Histogram Default Values**

The following table lists the Histogram factory default setup values.

<b>Command</b>	<b>Default Values</b>
HISTogram:BOXPcnt	30.0000,25.1000,70.0000,75.2000

Command	Default Values
HIStogram:DISplay	LINEAR
HIStogram:FUNcTion	HORIZONTAL
HIStogram:SIze	2.0000
HIStogram:SOUrce	CH1
HIStogram:STATE	0

### Horizontal Default Values

The following table lists the Horizontal factory default setup values.

Command	Default Values
HORizontal:ACQDURATION?	40 ns
HORizontal:ACQLENGTH?	1000
HORizontal[:MAIn]:DELay:MODE	0
HORizontal[:MAIn]:DELay:POSition	50.0000
HORizontal[:MAIn]:DELay:TIme	0.000000000000
HORizontal[:MAIn]:POSition	50.0000
HORizontal:DIgital:RECOrdlength?	1000
HORizontal:DIgital:SAMPLERate?	25 GS/s
HORizontal:FASTframe?	OFF
HORizontal:FASTframe:COUNt	2
HORizontal:FASTframe:MAXFRames?	1
HORizontal:MAIn:UNIts	"s"
HORizontal:MAIn:UNIts:STRing	"s"
HORizontal:MODE	AUTO
HORizontal:MODE:AUTO:LIMITrecorden	100000
HORizontal:MODE:RECOrdlength	≤2.5 GHz instruments: 1.0E+3
HORizontal:MODE:SAMPLERate	≥12.5 GHz instruments: 50 GS/s ≥3.5 GHz and ≤8 GHz instruments: 25.0 GS/s ≤3.5 GHz instruments: 10.0 GS/s
HORizontal:ROLL	AUTO

### Mask Default Values

The following table lists the Mask factory default setup values.

Command	Default Values
MASK:AUTOAdjust	10
MASK:AUTOAdjust:HDELTA	10
MASK:AUTOAdjust:VDELTA	10
MASK:AUTOSet:AUTOAdjust	1

Command	Default Values
MASK:AUTOSet:HPOS	1
MASK:AUTOSet:HSCAle	1
MASK:AUTOSet:MODe	MANUAL
MASK:AUTOSet:OFFSEtAdj	1
MASK:AUTOSet:STANdard	NONE
MASK:AUTOSet:TRIGger	1
MASK:AUTOSet:VPOS	1
MASK:AUTOSet:VSCAle	1
MASK:COUnT:STATE	0
MASK:DISPlay	1
MASK:FILTer	0
MASK:HIGHLIGHtHits	1
MASK:INVert	0
MASK:LOCK	1
MASK:MARgin:PERCent	5.0000
MASK:MARgin:STATE	0
MASK:MASKPRE:AMPlitude	100.0000E-3
MASK:MASKPRE:HSCAle	200.0000E-9
MASK:MASKPRE:HTRIGPOS	500.0000E-3
MASK:MASKPRE:PATTERNBITS	1
MASK:MASKPRE:PRESAMPBITS	0
MASK:MASKPRE:RECOrdlength	5000
MASK:MASKPRE:TRIGTOSAMP	0.0000
MASK:MASKPRE:VOFFSet	0.0000
MASK:MASKPRE:VPOS	0.0000
MASK:MASKPRE:VSCAle	1.0000E-3
MASK:MASKPRE:WIDth	0.0000
MASK:POLarity	POSITIVE
MASK:SOUrce	CH1
MASK:STANdard	NONE
MASK:STOPOnviolation	0
MASK:TEST:BEEP:COMPLetion	0
MASK:TEST:BEEP:FAILure	0
MASK:TEST:DELay	0.0000
MASK:TEST:HARDCopy	0
MASK:TEST:LOG:FAILure	0
MASK:TEST:REPeat	0
MASK:TEST:SAMple:THReshold	1



Command	Default Values
MASK:TEST:SAVEWFM	0
MASK:TEST:SAVEWFM:FILENAME	"C:/TekScope/Waveforms/"
MASK:TEST:SRQ:COMPLetion	0
MASK:TEST:SRQ:FAILure	0
MASK:TEST:STATE	0
MASK:TEST:STOP:FAILure	0
MASK:TEST:THReshold	1
MASK:TEST:WAVEform	20
MASK:USER:AMPLitude	100.0000E-3
MASK:USER:BITRate	1544000
MASK:USER:HSCALE	200.0000E-9
MASK:USER:HTRIGPOS	500.0000E-3
MASK:USER:LABel	"User Mask"
MASK:USER:PATTERNBITS	1
MASK:USER:PRESAMPBITS	0
MASK:USER:RECOrdlength	5000
MASK:USER:TRIGTOSAMP	0.0000
MASK:USER:VOFFSet	0.0000
MASK:USER:VPOS	0.0000
MASK:USER:VSCALE	1.0000E-3
MASK:USER:WIDth	0.0000

### Math Default Values

The following table lists the Math factory default setup values.

Command	Default Values
MATHVAR:VAR<x>	0.0000
MATH<x>:DEFine	" "
MATH<x>:LABel:NAME	""
MATH<x>:LABel:XPOS	1.0000
MATH<x>:LABel:YPOS	1 = 600.0000E-3 2 = 200.0000E-3 3 = -200.0000E-3 4 = -600.0000E-3
MATH<x>:NUMAVg	2
MATH<x>:SPECTral:CENTER	625.0000E+6
MATH<x>:SPECTral:GATEPOS	0.0000
MATH<x>:SPECTral:GATEWIDTH	2.0000E-6
MATH<x>:SPECTral:LOCK	0

Command	Default Values
MATH<x>:SPECTral:MAG	DB
MATH<x>:SPECTral:PHASE	DEGREES
MATH<x>:SPECTral:REFLevel	20.0000
MATH<x>:SPECTral:REFLEVELOffset	223.6000E-3
MATH<x>:SPECTral:RESBw	1000.0000E+3
MATH<x>:SPECTral:SPAN	1.2500E+9
MATH<x>:SPECTral:SUPPpress	-200.0000
MATH<x>:SPECTral:UNWRap	0
MATH<x>:SPECTral:WINdow	GAUSSIAN
MATH<x>:UNITString	"V"
MATH<x>:VERTical:POSition	0.0000
MATH<x>:VERTical:SCALE	1.0000

### Measurement Default Values

The following table lists the Measurement factory default setup values.

Command	Default Values
MEASUrement:GATing	OFF
MEASUrement:IMMed:DELay:DIREction	FORWARDS
MEASUrement:IMMed:DELay:EDGE[1]	RISE
MEASUrement:IMMed:DELay:EDGE2	RISE
MEASUrement:IMMed:METHod	HISTOGRAM
MEASUrement:IMMed:NOISe	HIGH
MEASUrement:IMMed:REFLevel:ABSolute:HIGH	0.0000
MEASUrement:IMMed:REFLevel:ABSolute:LOW	0.0000
MEASUrement:IMMed:REFLevel:ABSolute:MID<x>	0.0000
MEASUrement:IMMed:REFLevel:METHod	PERCENT
MEASUrement:IMMed:REFLevel:PERCent:HIGH	90.0000
MEASUrement:IMMed:REFLevel:PERCent:LOW	10.0000
MEASUrement:IMMed:REFLevel:PERCent:MID<x>	50.0000
MEASUrement:IMMed:SOUrce<x>	CH1
MEASUrement:IMMed:SOUrce<x>:SIGType	PULSE
MEASUrement:IMMed:TYPE	UNDEFINED
MEASUrement:MEAS<x>:DELay:DIREction	FORWARDS
MEASUrement:MEAS<x>:DELay:EDGE<x>	RISE
MEASUrement:MEAS<x>:METHod	HISTOGRAM
MEASUrement:MEAS<x>:NOISe	HIGH
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH	0.0000
MEASUrement:MEAS<x>:REFLevel:ABSolute:LOW	0.0000

Command	Default Values
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID<x>	0.0000
MEASUrement:IMMed:REFLevel:METHod	PERCENT
MEASUrement:MEAS<x>:REFLevel:PERCent:HIGH	0.0000
MEASUrement:MEAS<x>:REFLevel:PERCent:LOW	0.0000
MEASUrement:MEAS<x>:REFLevel:PERCent:MID<x>	0.0000
MEASUrement:MEAS<x>:SOUrce<x>	CH1
MEASUrement:MEAS<x>:SOUrce<x>:SIGType	PULSE
MEASUrement:MEAS<x>:STATE	0
MEASUrement:MEAS<x>:TYPe	UNDEFINED
MEASUrement:STATIstics:MODE	ALL
MEASUrement:STATIstics:WEIghting	32

### Miscellaneous Default Values

The following table lists the Miscellaneous factory default setup values.

Command	Default Values
AUXout:EDGE	FALLING
AUXout:SOUrce	ATRIGGER
CMDBatch	1 (ON)
HEADer	1 (ON)
LOCK	NONE
ROSc:TRACking	STABLE
VERBose	1 (ON)

### Save and Recall Default Settings

The following table lists the Save and Recall factory default setup values.

Command	Default Values
RECAll:SETUp:DESKew	OFF
SAVe:WAVEform:FILEFormat	INTERNAL

### Search and Mark Default Values

The following table lists the Search and Mark factory default setup values.

Command	Default Values
SEARCH:SEARCH<x>:TRIGger:A: DDRMemory:AUTOBitrate	AUTO
SEARCH:SEARCH<x>:TRIGger:A: DDRMemory:AUTORelevel	AUTO
SEARCH:SEARCH<x>:TRIGger:A: DDRMemory:BITRate	9.9000E+36

Command	Default Values
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:DATASource	CH1
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATa:HIGH	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATa:LOW	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:DATa:MID	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:HIGH	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:LOW	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:REFLevel:STROBE:MID	9.9000E+36
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STANdard	DDR3
SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:STROBEsource	CH2

**TekLink Default Values**

The following table lists the TekLink factory default setup values.

Command	Default Values
TEKLink:CONNection?	OFF
TEKLink:REFClk	INTERNAL
TRIGger:MULTiscope:ALIGN	DONE
TRIGger:MULTiscope:ALIGN:COMPLeted?	DONE
TRIGger:MULTiscope:LOGic	DONE

**Trigger Default Values**

The following table lists the Trigger factory default setup values.

Command	Default Values
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS}:PULSEForm	AMI = PLUSONE CMI = PLUSONE
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS}:THReshold:HIGH	AMI = 0.0000
TRIGger:A:COMMunication: {AMI HDB3 B3ZS B6ZS B8ZS}:THReshold:LOW	AMI = 0.0000
TRIGger:A:COMMunication:BITRate	1544000
TRIGger:A:COMMunication:CLOCK:POLarity	RISE
TRIGger:A:COMMunication:CODE	NRZ
TRIGger:A:COMMunication:SOUrce	CH1

Command	Default Values
TRIGger:A:COMMunication:SOURce:TYPe	DATA
TRIGger:A:COMMunication:STANdard	CUSTOM
TRIGger:{A B}:EDGE:COUPling	DC
TRIGger:{A B}:EDGE:SLOpe	RISE
TRIGger:{A B}:EDGE:SOURce	A = CH1 B = CH1
TRIGger:A:HOLDoff:BY	DEFAULT
TRIGger:A:HOLDoff:TIME	1.5000E-6
TRIGger:{A B}:LEVel	0.0000
TRIGger:{A B}:LOGIc:CLAss	PATTERN
TRIGger:{A B}:LOGIc:FUNcTION	AND
TRIGger:{A B}:LOGIc:INPut:CH<x>	CH1 = HIGH CH2 = X CH3 = X
TRIGger:{A B}:LOGIc:PATtern:INPut:CH<x>	CH4 = X
TRIGger:{A B}:LOGIc:PATtern:WHEn	TRUE
TRIGger:{A B}:LOGIc:PATtern:WHEn:LESSLimit	5.0000E-12
TRIGger:{A B}:LOGIc:PATtern:WHEn:MORELimit	5.0000E-12
TRIGger:{A B}:LOGIc:SETHold:CLOCK:EDGE	RISE
TRIGger:{A B}:LOGIc:SETHold:CLOCK:SOURce	CH2
TRIGger:{A B}:LOGIc:SETHold:CLOCK:THReshold	1.2000
TRIGger:{A B}:LOGIc:SETHold:DATA:SOURce	CH1
TRIGger:{A B}:LOGIc:SETHold:DATA:THReshold	1.2000
TRIGger:{A B}:LOGIc:SETHold:HOLDTime	2.0000E-9
TRIGger:{A B}:LOGIc:SETHold:SETTime	1.0000E-9
TRIGger:{A B}:LOGIc:STATE:INPut:CH<x>	CH4 = RISE
TRIGger:{A B}:LOGIc:STATE:WHEn	TRUE
TRIGger:{A B}:LOGIc:THReshold:CH<x>	CH4 = 1.2000
TRIGger:A:MODE	AUTO
TRIGger:{A B}:PULse:CLAss	GLITCH
TRIGger:{A B}:PULse:GLItch:POLarity	POSITIVE
TRIGger:{A B}:PULse:GLItch:TRIGIF	ACCEPT
TRIGger:{A B}:PULse:GLItch:WIDth	2.0000E-9
TRIGger:{A B}:PULse:RUNT:POLarity	POSITIVE
TRIGger:{A B}:PULse:RUNT:THReshold:HIGh	1.2000
TRIGger:{A B}:PULse:RUNT:THReshold:LOW	800.0000E-3
TRIGger:{A B}:PULse:RUNT:WHEn	OCCURS
TRIGger:{A B}:PULse:RUNT:WIDth	2.0000E-9

Command	Default Values
TRIGger:{A B}:PULse:SOUrce	CH1
TRIGger:{A B}:PULse:TIMEOut:POLarity	STAYSHIGH
TRIGger:{A B}:PULse:TIMEOut:TIME	2.0000E-9
TRIGger:{A B}:PULse:TRANSition:DELTAime	2.0000E-9
TRIGger:{A B}:PULse:TRANSition:POLarity	POSITIVE
TRIGger:{A B}:PULse:TRANSition:THReshold:HIGH	1.2000
TRIGger:{A B}:PULse:TRANSition:THReshold:LOW	800.0000E-3
TRIGger:{A B}:PULse:TRANSition:WHEn	FASTER THAN
TRIGger:{A B}:PULse:WIDth:HIGHLimit	2.0000E-9
TRIGger:{A B}:PULse:WIDth:LOWLimit	2.0000E-9
TRIGger:{A B}:PULse:WIDth:POLarity	POSITIVE
TRIGger:{A B}:PULse:WIDth:WHEn	WITHIN
TRIGger:{A B}:PULse:WINDow:THReshold:HIGH	1.2000
TRIGger:{A B}:PULse:WINDow:THReshold:LOW	800.0000E-3
TRIGger:{A B}:PULse:WINDow:WIDTh	2.0000E-9
TRIGger:{A B}:PULse:WINDow:POLarity	EITHER
TRIGger:{A B}:PULse:WINDow:POLarity:CH<x>	EITHER
TRIGger:{A B}:TYPE	EDGE
TRIGger:A:VIDeo:CUSTom:FORMat	INTERLACED
TRIGger:A:VIDeo:FIELD	ALLFIELDS
TRIGger:A:VIDeo:HOLdoff:FIELD	0.0000
TRIGger:A:VIDeo:LINE	1
TRIGger:A:VIDeo:POLarity	NORMAL
TRIGger:A:VIDeo:SCAN	RATE1
TRIGger:A:VIDeo:SOUrce	CH1
TRIGger:A:VIDeo:STANdard	NTSC
TRIGger:B:BY	EVENTS
TRIGger:B:EVENTS:COUNT	2
TRIGger:B:STATE	0
TRIGger:B:TIME	16.0000E-9

### Vertical Default Values

The following table lists the Vertical factory default setup values.

Command	Default Values
CH<x>:BANdwidth	500.0000E+6
CH<x>:COUPling	DC
CH<x>:DESKew	0.0000
	0

Command	Default Values
CH<x>:LABel:NAME	""
CH<x>:LABel:XPOS	CH1 = 1.0000
CH<x>:LABel:YPOS	CH1 = 2.2000
	CH2 = 1.8000
	CH3 = 1.4000
	CH4 = 1.000
CH<x>:OFFSet	0.0000
CH<x>:POSition	0.0000
CH<x>:PROBECOntrOl	AUTO
CH<x>:PROBEFunc:EXTAtten	1.0000
CH<x>:PROBEFunc:EXTUnits	"V"
CH<x>:SCAle	100.0000E-3
CH<x>:TERmination	1.0000E+6
REF<x>:LABel:NAME	""
REF<x>:LABel:XPOS	1.0000
REF<x>:LABel:YPOS	-1.0000
SElect:<wfm>	CH1 = 1
	CH2 = 0
	CH3 = 0
	CH4 = 0
SElect:CONTROl	CH1

### Waveform Default Values

The following table lists the Waveform factory default setup values.

**NOTE.** These are not set to factory values when a default setup is executed. :DATA INIT is used to reset these values.

Command	Default Values
DATA:DESTination	REF1
DATA:ENCdg	RIBINARY
DATA:SOUrce	CH1
DATA:STARt	1
DATA:STOP	1000

### Zoom Default Values

The following table lists the Zoom factory default setup values.

Command	Default Values
ZOOM:GRAticule:SIZE	80

<b>Command</b>	<b>Default Values</b>
ZOOm:MODE	0
ZOOm:SCROLL:DIREction	STOP
ZOOm:SCROLL:LOCK	0
ZOOm:SCROLL:SPEED	1
ZOOm:ZOOM<x>:<wfm>:DISplay	0
ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition	Zoom1 50.0000 Zoom2 30.0000 Zoom3 70.0000 Zoom4 90.0000
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAle	5
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition	0.0000
ZOOm:ZOOM<x>:<wfm>:VERTical:SCAle	1.0000
ZOOm:ZOOM<x>:SCROLLLock	1
ZOOm:STATE	1



---

# Appendix D: GPIB Interface Specifications

This topic describes details of the GPIB remote interface of the instrument. Normally, you will not need this information to use the instrument, but this information may be useful if you are connecting to controllers with unusual configurations.

## Interface Messages

The following table shows the standard interface messages that the instrument supports.

**Table D-1: Standard Interface Messages**

<b>Message</b>	<b>Interface</b>
DCL	Yes
GET	Yes
GTL	Yes
LLO	Yes
PPC	No
PPD	No
PPE	No
PPU	No
SDC	Yes
SPD	Yes
SPE	Yes
TCT	No
UNL	Yes
UNT	Yes
Listen Addresses	Yes
Talk Addresses	Yes

## GPIB Functions

The following table lists the interface functions and electrical functions subsets supported by this instrument with brief descriptions of each.

**Table D-2: Interface Functions**

Interface Function	Subset	Meaning
Acceptor Handshake	AH1	The instrument can receive multiline messages across the interface from other devices.
Controller	C1	Controller capability, the instrument can control other devices.
Device Clear	DC1	The instrument can respond to both the DCL (Device Clear) interface message and to the Selected Device Clear (SDC) interface message when the instrument is listen-addressed.
Device Trigger	DT1	Device Trigger capability, the instrument does respond to the GET (Group Execute Trigger) interface message.
Electrical	E2	The instrument uses tri-state buffers, which are optimal for high-speed data transfer.
Listener	L4	The instrument becomes a listener when it detects the listen address being sent over the bus with the ATN line asserted. The instrument ceases to be a listener and becomes a talker when it detects the talk address being sent over the bus with the ATN line asserted.
Parallel Poll	PP0	No Parallel Poll Capability, the instrument does not respond to PPC (Parallel Poll Configure), PPO (Parallel POI Disable), PPE (Parallel Poll Enable), or PPU (Parallel Poll Unconfigure) interface messages, nor does it send a status message when the ATN and EOI lines are asserted simultaneously.
Remote/Local	RL1	The instrument can respond to both GTL (Go To Local) and LLO (Local Lock Out) interface messages.
Service Request	SR1	The instrument can assert the SRQ line to notify the controller in charge that it requires service.
Source Handshake	SH1	The instrument can initiate multiline messages to send across the interface to other devices.
Talker	T5	The instrument becomes a talker when it detects the talk address being sent over the bus with the ATN line asserted. The instrument ceases to be a talker and becomes a listener when it detects the listen address being sent over the bus with the ATN line asserted. The instrument also ceases to be a talker when it detects the talk address of another device being sent over the data lines with ATN asserted.

---

# 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 on the GPIB interface bus. 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.

## **GPIB**

An acronym for General Purpose Interface Bus, the common name for the interface system defined in IEEE Std-488.

## **IEEE**

An acronym for the Institute for Electrical and Electronic Engineers.

**TEKSecure**

A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.

**Serial Poll**

A device on the GPIB bus can request service from the GPIB Controller by asserting the GPIB SRQ line (a Hardware line that is only present on the GPIB communications bus). When a controller acknowledges the SRQ, it “serial polls” each open device on the bus to determine which device on the bus requested service. Any device requesting service returns a status byte with bit 6 set and then unasserts the SRQ line. Devices not requiring service return a status byte with bit 6 cleared.

# Index

## A

ACQUIRE:INTERPEightbit, 2-93  
ACQUIRE:NUMFRAMESACQUIRED?, 2-97  
ACQUIRE:SAMPLINGmode, 2-98  
ACQUIRE?, 2-93  
ACQUIRE:MODE, 2-94  
ACQUIRE:NUMACQ?, 2-95  
ACQUIRE:NUMAVG, 2-96  
ACQUIRE:NUMENV, 2-96  
ACQUIRE:NUMSAMPLES, 2-98  
ACQUIRE:STATE, 2-99  
ACQUIRE:STOPAFTER, 2-100  
ALIAS, 2-100  
ALIAS:CATALOG?, 2-101  
ALIAS:DEFINE, 2-101  
ALIAS:DELETE, 2-102  
ALIAS:DELETE:ALL, 2-103  
ALIAS:DELETE:NAME, 2-103  
ALIAS:STATE, 2-103  
ALLEV?, 2-104  
ALLOcate:WAVEform:REF<x>?, 2-104  
AUTOSet, 2-105  
AUXIn:PRObe:AUTOZero, 2-107  
AUXIn:PRObe:DEGAUSS:STATE?, 2-108  
AUXIn:PRObe:FORCEDRange, 2-108  
AUXIn:PRObe:ID:SERnumber?, 2-109  
AUXIn:PRObe:ID:TYPE?, 2-109  
AUXIn:PRObe:INPUTMode, 2-110  
AUXIn:PRObe:INPUTMode:AOFFSet, 2-110  
AUXIn:PRObe:INPUTMode:BOFFSet, 2-111  
AUXIn:PRObe:INPUTMode:CMOFFSet, 2-111  
AUXIn:PRObe:INPUTMode:DMOFFSet, 2-112  
AUXIn:PRObe:RESistance?, 2-113  
AUXIn:PROBEFunc:EXTatten, 2-115  
AUXIn:PROBEFunc:EXTDBatten, 2-116  
AUXIn:PROBEFunc:EXTUnits, 2-117  
AUXIn:BANDwidth, 2-105  
AUXIn:COUPLing, 2-106  
AUXIn:OFFSet, 2-106  
AUXIn:PRObe:DEGAUSS, 2-107  
AUXIn:PRObe:GAIN?, 2-109  
AUXIn:PRObe:RANge, 2-112  
AUXIn:PRObe:SET, 2-113  
AUXIn:PRObe:SIGNAL, 2-114

AUXIn:PRObe:TIPtype, 2-115  
AUXIn:PRObe:UNITS?, 2-115  
AUXout?, 2-118  
AUXout:EDGE, 2-118  
AUXout:SOURce, 2-118

## B

BELI, 2-119  
BUS:B<x>:DISplay:DECOde:FILE, 2-119  
BUS:B<x>:DISplay:DECOde:STATe, 2-120  
BUS:B<x>:DISplay:FORMat, 2-121  
BUS:B<x>:DISplay:TYPE, 2-121  
BUS:B<x>:I2C:CLOCK:SOURce, 2-122  
BUS:B<x>:I2C:DATA:SOURce, 2-123  
BUS:B<x>:I2C:RWINADDR, 2-123  
BUS:B<x>:PARallel:CLOCK:EDGE, 2-125  
BUS:B<x>:PARallel:CLOCK:SOURce, 2-125  
BUS:B<x>:PARallel:ISCLOCKED, 2-126  
BUS:B<x>:PARallel:SOURCES, 2-127  
BUS:B<x>:POSition, 2-128  
BUS:B<x>:SPI:BITOrder, 2-128  
BUS:B<x>:SPI:CLOCK:POLarity, 2-129  
BUS:B<x>:SPI:CLOCK:SOURce, 2-129  
BUS:B<x>:SPI:DATA:DELay, 2-130  
BUS:B<x>:SPI:DATA:POLarity, 2-131  
BUS:B<x>:SPI:DATA:SIZE, 2-131  
BUS:B<x>:SPI:DATA:SOURce, 2-132  
BUS:B<x>:SPI:SELEct:POLarity, 2-133  
BUS:B<x>:SPI:SELEct:SOURce, 2-133  
BUS:B<x>:LABel, 2-124  
BUS:B<x>:TYPE, 2-134  
BUSY?, 2-134

## C

\*CAL?, 2-135  
CALibrate:CALProbe:CH<x>?, 2-136  
CALibrate:INTERNAL:STARt, 2-137  
CALibrate:INTERNAL:STATus?, 2-137  
CALibrate:PRObestate:CH<x>?, 2-138  
CALibrate:RESults:SPC?, 2-139  
CALibrate?, 2-135  
CALibrate:INTERNAL, 2-136  
CALibrate:RESults?, 2-139

- CH<x>:BANdwidth:ENHanced, 2-141
  - CH<x>:BANdwidth:ENHanced:APPLYtoall, 2-141
  - CH<x>:BANdwidth:ENHanced:FORCe, 2-142
  - CH<x>:BANdwidth:ENHanced:STATE?, 2-143
  - CH<x>:ICAPture:SOUrce, 2-145
  - CH<x>:ICAPture:STATE, 2-145
  - CH<x>:PRObe:AUTOZero, 2-150
  - CH<x>:PRObe:DEGAUSS, 2-150
  - CH<x>:PRObe:DEGAUSS:STATE?, 2-150
  - CH<x>:PRObe:FORCEDRange, 2-151
  - CH<x>:PRObe:GAIN?, 2-151
  - CH<x>:PRObe:ID:SERnumber?, 2-152
  - CH<x>:PRObe:ID:TYPe?, 2-153
  - CH<x>:PRObe:INPUTMode, 2-153
  - CH<x>:PRObe:INPUTMode:AOFFSet, 2-154
  - CH<x>:PRObe:INPUTMode:BOFFSet, 2-154
  - CH<x>:PRObe:INPUTMode:CMOFFSet, 2-155
  - CH<x>:PRObe:INPUTMode:DMOFFSet, 2-155
  - CH<x>:PRObe:RANge, 2-156
  - CH<x>:PRObe:RESistance?, 2-157
  - CH<x>:PRObe:SIGnal, 2-158
  - CH<x>:PRObe:TIPtype, 2-159
  - CH<x>:PRObe:UNIts?, 2-159
  - CH<x>:PROBECOntrol, 2-160
  - CH<x>:PROBEFunc:EXTAtten, 2-161
  - CH<x>:PROBEFunc:EXTDBatten, 2-161
  - CH<x>:PROBEFunc:EXTUnits, 2-162
  - CH<x>:TERmination, 2-164
  - CH<x>?, 2-140
  - CH<x>:BANdwidth, 2-140
  - CH<x>:COUPling, 2-143
  - CH<x>:DESKew, 2-144
  - CH<x>:LABel:NAME, 2-146
  - CH<x>:LABel:XPOS, 2-146
  - CH<x>:LABel:YPOS, 2-147
  - CH<x>:OFFSet, 2-147
  - CH<x>:POSition, 2-148
  - CH<x>:PRObe?, 2-149
  - CH<x>:PRObe:ID?, 2-152
  - CH<x>:PRObe:SET, 2-157
  - CH<x>:PROBECal?, 2-160
  - CH<x>:SCAle, 2-163
  - CH<x>:THRESHold, 2-164
  - \*CLS, 2-165
  - CMDBatch, 2-165
  - Command Groups, 2-11
  - CQ<x>:THRESHold, 2-166
  - CURSor:HBArS:POSITION<x>, 2-169
  - CURSor:SCREEN:XPOSITION<x>, 2-171
  - CURSor:SCREEN:YPOSITION<x>, 2-172
  - CURSor:VBArS:POS<x>, 2-175
  - CURSor:VBArS:POSITION<x>, 2-175
  - CURSor:WAVEform:HDELTA?, 2-177
  - CURSor:WAVEform:HPOS<x>?, 2-177
  - CURSor:WAVEform:POSition<x>, 2-178
  - CURSor:WAVEform:STYle, 2-178
  - CURSor:WAVEform:UNIts, 2-179
  - CURSor:WAVEform:VDELTA?, 2-179
  - CURSor:XY:PRODELta?, 2-180
  - CURSor:XY:PRODUCT<x>?, 2-181
  - CURSor:XY:RADIUS<x>?, 2-181
  - CURSor:XY:RATIO<x>?, 2-182
  - CURSor:XY:RECTX<x>, 2-183
  - CURSor:XY:RECTY<x>, 2-184
  - CURSor:XY:THETA<x>?, 2-185
  - CURSor?, 2-167
  - CURSor:FUNcTION, 2-167
  - CURSor:HBArS?, 2-168
  - CURSor:HBArS:DELta?, 2-168
  - CURSor:HBArS:UNIts?, 2-169
  - CURSor:LINEStYle, 2-170
  - CURSor:MODE, 2-170
  - CURSor:SCREEN:STYle, 2-171
  - CURSor:SOUrce<x>, 2-172
  - CURSor:STATE, 2-173
  - CURSor:VBArS, 2-174
  - CURSor:VBArS:DELta?, 2-174
  - CURSor:VBArS:UNIts, 2-176
  - CURSor:WAVEform, 2-176
  - CURSor:XY?, 2-180
  - CURSor:XY:RATDELta?, 2-181
  - CURSor:XY:RDELta?, 2-182
  - CURSor:XY:READOUT, 2-183
  - CURSor:XY:THDELta?, 2-184
  - CURSor:XY:XDELta?, 2-185
  - CURSor:XY:YDELta?, 2-186
  - CURVe, 2-186
  - CURVENext?, 2-188
  - CURVStream, 2-188
- ## D
- D<x>:PROBE:ID:SERnumber?, 2-240
  - D<x>:PROBE:ID:TYPe?, 2-240
  - D<x>:LABEL, 2-238

- D<x>:POSition, 2-239  
 D<x>:THRESHold, 2-240  
 DATa, 2-189  
 DATa:DESTination, 2-190  
 DATa:ENCdg, 2-190  
 DATa:FRAMEStArt, 2-193  
 DATa:FRAMEStOP, 2-194  
 DATa:SOUrce, 2-194  
 DATa:StArt, 2-195  
 DATa:StOP, 2-196  
 DATa:SYNCSOUrces, 2-196  
 DATE, 2-197  
 \*DDT, 2-198  
 DELEte:SEtUp, 2-198  
 DELEte:WAVEform, 2-199  
 DESE, 2-199  
 DIAg:RESults:VERBose?, 2-209  
 DIAg:CONtROL:HALT, 2-200  
 DIAg:CONtROL:LOOP, 2-201  
 DIAg:EXECUTE, 2-202  
 DIAg:FAILURES:CLear, 2-202  
 DIAg:ITEM?, 2-203  
 DIAg:ITEM:FAILURES?, 2-203  
 DIAg:ITEM:NAME?, 2-204  
 DIAg:ITEM:RESult?, 2-204  
 DIAg:ITEM:SUBITEMS?, 2-205  
 DIAg:LEVEL, 2-205  
 DIAg:LOOPS?, 2-206  
 DIAg:NAME?, 2-206  
 DIAg:NAME:AREA?, 2-207  
 DIAg:NAME:SUBSYS?, 2-207  
 DIAg:NAME:TEST?, 2-208  
 DIAg:NUMITEMS?, 2-208  
 DIAg:RESults?, 2-208  
 DIAg:SELEct:ALL, 2-209  
 DIAg:SELEct:AREA, 2-210  
 DIAg:SELEct:LAST, 2-210  
 DIAg:SELEct:SUBSYS, 2-211  
 DIAg:SELEct:TEST, 2-212  
 DIAg:StATE, 2-212  
 DIAg:StOP, 2-213  
 DISPlay:COLOr:MATHCOLOr, 2-215  
 DISPlay:COLOr:PALEtte:IMAGeView, 2-216  
 DISPlay:COLOr:PALEtte:RECORDView, 2-217  
 DISPlay:COLOr:PALEtte:USER, 2-218  
 DISPlay:COLOr:PALEtte:USER:CARet, 2-218  
 DISPlay:COLOr:PALEtte:USER:CH<x>, 2-219  
 DISPlay:COLOr:PALEtte:USER:GRAticule, 2-219  
 DISPlay:COLOr:PALEtte:USER:HIStogram, 2-220  
 DISPlay:COLOr:PALEtte:USER:MASK, 2-220  
 DISPlay:COLOr:PALEtte:USER:  
     MASKHighLight, 2-221  
 DISPlay:COLOr:PALEtte:USER:MATH<x>, 2-222  
 DISPlay:COLOr:PALEtte:USER:REF<x>, 2-222  
 DISPlay:COLOr:PALEtte:USER:WAVEform, 2-223  
 DISPlay:COLOr:REFCOLOr, 2-223  
 DISPlay:DIGital:HEIGHt, 2-224  
 DISPlay:INTENSITy:WAVEform:  
     IMAGeView, 2-227  
 DISPlay:INTENSITy:WAVEform:  
     RECORDView, 2-228  
 DISPlay:PERStence:RESEt, 2-229  
 DISPlay:SCREENtExt:LABel<x>:  
     FONTCOLOr, 2-230  
 DISPlay:SCREENtExt:LABel<x>:  
     FONtNAME, 2-231  
 DISPlay:SCREENtExt:LABel<x>:FONtStIze, 2-231  
 DISPlay:SCREENtExt:LABel<x>:  
     FONtStYle, 2-232  
 DISPlay:SCREENtExt:LABel<x>:NAME, 2-232  
 DISPlay:SCREENtExt:LABel<x>:StATE, 2-233  
 DISPlay:SCREENtExt:LABel<x>:XPOS, 2-233  
 DISPlay:SCREENtExt:LABel<x>:YPOS, 2-234  
 DISPlay:SCREENtExt:LABel<x>?, 2-230  
 DISPlay:SCREENtExt:StATE, 2-234  
 DISPlay?, 2-214  
 DISPlay:CLOCK, 2-214  
 DISPlay:COLOr?, 2-215  
 DISPlay:FILTer, 2-225  
 DISPlay:FORMAt, 2-225  
 DISPlay:GRAticule, 2-226  
 DISPlay:INTENSITy?, 2-227  
 DISPlay:PERStence, 2-228  
 DISPlay:SCREENtExt?, 2-229  
 DISPlay:SHOWREmote, 2-235  
 DISPlay:StYle, 2-236  
 DISPlay:TRIGBar, 2-236  
 DISPlay:TRIGT, 2-237  
 DISPlay:VARpersist, 2-237  
 DISPlay:WAVEform, 2-238
- E**  
 EMail, 2-241  
 EMail:ATTempts, 2-241

EEmail:AUTHLogin, 2-242  
EEmail:AUTHPassword, 2-242  
EEmail:COUNT?, 2-243  
EEmail:FRom, 2-243  
EEmail:HOSTwanted, 2-244  
EEmail:IMAGe, 2-244  
EEmail:MASK, 2-245  
EEmail:MAXSize, 2-245  
EEmail:MEASUrement, 2-246  
EEmail:NUMEMails, 2-246  
EEmail:SMTPPort, 2-247  
EEmail:SMTPServer, 2-247  
EEmail:STATUS?, 2-248  
EEmail:TIMEOut, 2-248  
EEmail:TO, 2-249  
EEmail:TRIGger, 2-249  
EEmail:WAVEform, 2-250  
\*ESE, 2-250  
\*ESR?, 2-251  
EVENT?, 2-251  
EVMsg?, 2-252  
EVQty?, 2-252  
EXPort, 2-253  
EXPort:FILENAME, 2-253  
EXPort:FORMat, 2-254  
EXPort:PALEtte, 2-254  
EXPort:READOuts, 2-255  
EXPort:VIEW, 2-255

## F

FACTory, 2-256  
FASTAcq?, 2-257  
FASTAcq:HIACQRATE, 2-257  
FASTAcq:STATE, 2-258  
FILESystem:WRITEFile, 2-264  
FILESystem?, 2-258  
FILESystem:COPy, 2-259  
FILESystem:CWD, 2-259  
FILESystem:DELEte, 2-260  
FILESystem:DIR?, 2-261  
FILESystem:MKDir, 2-261  
FILESystem:PRInt, 2-262  
FILESystem:READFile, 2-262  
FILESystem:REName, 2-263  
FILESystem:RMDir, 2-264

## H

HARDCopy, 2-265  
HARDCopy:FILENAME, 2-266  
HARDCopy:LAYout, 2-266  
HARDCopy:PALEtte, 2-267  
HARDCopy:PORT, 2-267  
HARDCopy:READOuts, 2-268  
HARDCopy:VIEW, 2-269  
HDR, 2-269  
HEADer, 2-270  
HIStoqram?, 2-271  
HIStoqram:BOX, 2-271  
HIStoqram:BOXPct, 2-272  
HIStoqram:COUNT, 2-273  
HIStoqram:DATA?, 2-273  
HIStoqram:DISplay, 2-274  
HIStoqram:FUNcTION, 2-275  
HIStoqram:MODE, 2-276  
HIStoqram:SIZE, 2-276  
HIStoqram:SOURce, 2-277  
HIStoqram:STATE, 2-278  
HORizontal:ACQDURATION?, 2-279  
HORizontal:ACQLENGTH?, 2-279  
HORizontal:DIGital:RECOrdlength?, 2-279  
HORizontal:DIGital:SAMPLERate?, 2-280  
HORizontal:DIVisions?, 2-280  
HORizontal:FASTframe:COUNt, 2-281  
HORizontal:FASTframe:MAXFRames?, 2-282  
HORizontal:FASTframe:MULTipleframes:  
    FRAMEStart:<wfm>, 2-282  
HORizontal:FASTframe:MULTipleframes:  
    MODE, 2-283  
HORizontal:FASTframe:MULTipleframes:  
    NUMFRames:<wfm>, 2-283  
HORizontal:FASTframe:REF:FRame, 2-284  
HORizontal:FASTframe:REF:SOURce, 2-285  
HORizontal:FASTframe:SELECTED:CH<x>, 2-285  
HORizontal:FASTframe:SELECTED:  
    MATH<x>, 2-286  
HORizontal:FASTframe:SELECTED:REF<x>, 2-286  
HORizontal:FASTframe:SELECTED:SOURce, 2-287  
HORizontal:FASTframe:SEQUence, 2-287  
HORizontal:FASTframe:STATE, 2-288  
HORizontal:FASTframe:SUMFrame, 2-288  
HORizontal:FASTframe:TIMEStamp:ALL:  
    <wfm>?, 2-289



HORizontal:FASTframe:TIMEStamp:BETween:  
     <wfm>?, 2-290  
 HORizontal:FASTframe:TIMEStamp:DELTA:  
     <wfm>?, 2-290  
 HORizontal:FASTframe:TIMEStamp:FRAME:  
     <wfm>?, 2-291  
 HORizontal:FASTframe:TIMEStamp:REF?, 2-292  
 HORizontal:FASTframe:TIMEStamp:SELECTED:  
     <wfm>?, 2-292  
 HORizontal:FASTframe:TRACk, 2-293  
 HORizontal:FASTframe:XZero:ALL:CH<x>?, 2-294  
 HORizontal:FASTframe:XZero:ALL:  
     REF<x>?, 2-294  
 HORizontal:FASTframe:XZero:FRame:  
     CH<x>?, 2-295  
 HORizontal:FASTframe:XZero:FRame:  
     REF<x>?, 2-295  
 HORizontal:FASTframe:XZero:REF?, 2-295  
 HORizontal:FASTframe:XZero:SELECTED:  
     CH<x>?, 2-296  
 HORizontal:FASTframe:XZero:SELECTED:  
     REF<x>?, 2-296  
 HORizontal:FASTframe?, 2-281  
 HORizontal:MAIn:INTERPRatio?, 2-299  
 HORizontal:MAIn:UNIts, 2-300  
 HORizontal:MAIn:UNIts:STRing, 2-300  
 HORizontal:MODE:AUTO:LIMITrecordlen, 2-302  
 HORizontal:MODE:RECOrdlength, 2-302  
 HORizontal:MODE:SAMPLERate, 2-302  
 HORizontal:MODE:SCAle, 2-303  
 HORizontal:TIMEStamp:CH<x>?, 2-305  
 HORizontal:TIMEStamp:REF<x>?, 2-305  
 HORizontal?, 2-278  
 HORizontal[:MAIn]:DELay:MODE, 2-297  
 HORizontal[:MAIn]:DELay:POSition, 2-298  
 HORizontal[:MAIn]:DELay:TIME, 2-298  
 HORizontal[:MAIn]:POSition, 2-299  
 HORizontal:MAIn?, 2-297  
 HORizontal:MODE, 2-301  
 HORizontal:POSition, 2-303  
 HORizontal:ROLL, 2-304

## I

ID?, 2-305  
 \*IDN?, 2-306

## L

LIMit:COMpare:CH<x>, 2-308  
 LIMit:COMpare:MATH<x>, 2-308  
 LIMit:COMpare:REF<x>, 2-309  
 LIMit:HIGHLIGHTHits:RESet, 2-312  
 LIMit:SAVEWFM:FILEName, 2-314  
 LIMit:STOPOnviolation, 2-316  
 LIMit:TEMPlate:DESTination, 2-317  
 LIMit:TEMPlate:SOUrce, 2-317  
 LIMit:TEMPlate:STORe, 2-318  
 LIMit:TEMPlate:TOLerance:HORizontal, 2-319  
 LIMit:TEMPlate:TOLerance:VERTical, 2-319  
 LIMit?, 2-306  
 LIMit:BEEP, 2-307  
 LIMit:COMpare, 2-307  
 LIMit:EMail, 2-309  
 LIMit:HARDCopy, 2-310  
 LIMit:HIGHLIGHTHits, 2-311  
 LIMit:LOCK, 2-312  
 LIMit:LOG, 2-313  
 LIMit:SAVEWFM, 2-313  
 LIMit:SRQ, 2-315  
 LIMit:STATE, 2-315  
 LIMit:STATus?, 2-316  
 LOCK, 2-320  
 \*LRN?, 2-321

## M

MARK, 2-321  
 MARK:SELECTED:FOCUS?, 2-324  
 MARK:SELECTED:MARKSINCOLUMN?, 2-325  
 MARK:SELECTED:OWNer?, 2-325  
 MARK:SELECTED:SOUrce?, 2-325  
 MARK:SELECTED:STARt?, 2-326  
 MARK:SELECTED:ZOOM:POSition?, 2-326  
 MARK:SELECTED:ZOOM:SCAle?, 2-327  
 MARK:CREATE, 2-322  
 MARK:DELEte, 2-323  
 MARK:FREE?, 2-323  
 MARK:SELECTED:END?, 2-323  
 MARK:SELECTED:LABel, 2-324  
 MARK:SELECTED:STATE, 2-326  
 MARK:TOTal?, 2-327  
 MASK:AUTOAdjust:HDELTA, 2-328  
 MASK:AUTOAdjust:VDELTA, 2-329  
 MASK:AUTOSet:AUTOAdjust, 2-329  
 MASK:AUTOSet:OFFSETAdj, 2-332

- MASK:AUTOSet:STANdard, 2-332
- MASK:AUTOSet:TRIGger, 2-338
- MASK:AUTOSet:USER:ONE, 2-339
- MASK:AUTOSet:USER:TYPE, 2-339
- MASK:AUTOSet:USER:ZERo, 2-340
- MASK:COUNT:FAILURES?, 2-342
- MASK:COUNT:SEG<m>:HITS?, 2-343
- MASK:COUNT:VIOLATIONS?, 2-345
- MASK:COUNT:WAVEFORMS?, 2-345
- MASK:MASKPRE:AMPLitude, 2-349
- MASK:MASKPRE:HTRIGPOS, 2-351
- MASK:MASKPRE:PATTERNBITS, 2-351
- MASK:MASKPRE:PRESAMPBITS, 2-352
- MASK:MASKPRE:RECOrdlength, 2-352
- MASK:MASKPRE:TRIGTOSAMP, 2-353
- MASK:MASKPRE:VOFFSet, 2-353
- MASK:SEG<m>:NR\_Pt?, 2-357
- MASK:SEG<m>:POINTS, 2-357
- MASK:STOPOnviolation, 2-365
- MASK:TEST:AUX:COMPLetion, 2-366
- MASK:TEST:AUX:FAILure, 2-366
- MASK:TEST:BEEP:COMPLetion, 2-367
- MASK:TEST:BEEP:FAILure, 2-367
- MASK:TEST:LOG:FAILure, 2-369
- MASK:TEST:SAMple:THReshold, 2-371
- MASK:TEST:SAVEWFM:FILEName, 2-372
- MASK:TEST:SRQ:COMPLetion, 2-373
- MASK:TEST:SRQ:FAILure, 2-373
- MASK:TEST:STOP:FAILure, 2-375
- MASK:USER:PATTERNBITS, 2-379
- MASK:USER:PRESAMPBITS, 2-380
- MASK:USER:RECOrdlength, 2-380
- MASK:USER:SEG<m>:NR\_Pt?, 2-381
- MASK:USER:SEG<m>:POINTS, 2-381
- MASK:USER:TRIGTOSAMP, 2-382
- MASK?, 2-327
- MASK:AUTOAdjust, 2-328
- MASK:AUTOSet:HPOS, 2-330
- MASK:AUTOSet:HSCALE, 2-331
- MASK:AUTOSet:MODE, 2-331
- MASK:AUTOSet:VPOS, 2-340
- MASK:AUTOSet:VSCALE, 2-341
- MASK:COpy:USER, 2-341
- MASK:COUNT, 2-342
- MASK:COUNT:HITS?, 2-342
- MASK:COUNT:STATE, 2-343
- MASK:COUNT:TESTS?, 2-344
- MASK:COUNT:TOTal?, 2-344
- MASK:DISPlay, 2-346
- MASK:FILTer, 2-346
- MASK:HIGHLIGHTHits, 2-347
- MASK:INVert, 2-347
- MASK:LOCK, 2-348
- MASK:MARgin:PERCent, 2-348
- MASK:MARgin:STATE, 2-349
- MASK:MASKPRE:HSCALE, 2-350
- MASK:MASKPRE:VPOS, 2-354
- MASK:MASKPRE:VSCALE, 2-354
- MASK:MASKPRE:WIDth, 2-355
- MASK:POLarity, 2-356
- MASK:SEG<m>, 2-356
- MASK:SOUrce, 2-358
- MASK:STANdard, 2-358
- MASK:TEST:DELAy, 2-368
- MASK:TEST:HARDCopy, 2-368
- MASK:TEST:REPeat, 2-370
- MASK:TEST:SAMple, 2-370
- MASK:TEST:SAVEWFM, 2-371
- MASK:TEST:STATE, 2-374
- MASK:TEST:STATUS?, 2-374
- MASK:TEST:THReshold, 2-375
- MASK:TEST:WAVEform, 2-376
- MASK:USER:AMPLitude, 2-376
- MASK:USER:BITRate, 2-377
- MASK:USER:HSCALE, 2-377
- MASK:USER:HTRIGPOS, 2-378
- MASK:USER:LABel, 2-379
- MASK:USER:SEG<m>, 2-381
- MASK:USER:VOFFSet, 2-383
- MASK:USER:VPOS, 2-383
- MASK:USER:VSCALE, 2-384
- MASK:USER:WIDth, 2-384
- MATH<x>:FILTer:MODE, 2-388
- MATH<x>:FILTer:RISetime, 2-389
- MATH<x>:LABel:NAME, 2-389
- MATH<x>:LABel:XPOS, 2-390
- MATH<x>:LABel:YPOS, 2-390
- MATH<x>:SPECTral:CENTER, 2-392
- MATH<x>:SPECTral:GATEPOS, 2-393
- MATH<x>:SPECTral:GATEWIDTH, 2-393
- MATH<x>:SPECTral:LOCK, 2-394
- MATH<x>:SPECTral:MAG, 2-395
- MATH<x>:SPECTral:PHASE, 2-396
- MATH<x>:SPECTral:REFLevel, 2-396

- MATH<x>:SPECTral:REFLEVELOffset, 2-397  
 MATH<x>:SPECTral:RESBw, 2-398  
 MATH<x>:SPECTral:SPAN, 2-398  
 MATH<x>:SPECTral:SUPPress, 2-399  
 MATH<x>:SPECTral:UNWRap, 2-400  
 MATH<x>:SPECTral:WINDow, 2-400  
 MATH<x>:SPECTral?, 2-392  
 MATH<x>:THRESHold, 2-401  
 MATH<x>:UNITString, 2-402  
 MATH<x>:VERTical:POSition, 2-402  
 MATH<x>:VERTical:SCALE, 2-403  
 MATH<x>?, 2-387  
 MATH<x>:DEFine, 2-387  
 MATH<x>:NUMAVg, 2-391  
 MATHArbflt<x>:FILEpath, 2-385  
 MATHArbflt<x>:READFile, 2-385  
 MATHVAR?, 2-386  
 MATHVAR:VAR<x>, 2-386  
 MEASUrement:ANNOtation:IMMEDSTate, 2-405  
 MEASUrement:ANNOtation:NUMX?, 2-406  
 MEASUrement:ANNOtation:NUMY?, 2-406  
 MEASUrement:ANNOtation:STATE, 2-406  
 MEASUrement:ANNOtation:TYPE, 2-407  
 MEASUrement:ANNOtation:X<x>?, 2-407  
 MEASUrement:ANNOtation:Y<x>?, 2-408  
 MEASUrement:IMMed:DELay:DIREction, 2-410  
 MEASUrement:IMMed:DELay:EDGE[1], 2-410  
 MEASUrement:IMMed:DELay:EDGE2, 2-411  
 MEASUrement:IMMed:DELay?, 2-409  
 MEASUrement:IMMed:METHod, 2-412  
 MEASUrement:IMMed:NOISe, 2-412  
 MEASUrement:IMMed:REFLevel:ABSolute:  
   HIGH, 2-413  
 MEASUrement:IMMed:REFLevel:ABSolute:  
   LOW, 2-414  
 MEASUrement:IMMed:REFLevel:ABSolute:  
   MID<x>, 2-414  
 MEASUrement:IMMed:REFLevel:METHod, 2-415  
 MEASUrement:IMMed:REFLevel:PERCent:  
   HIGH, 2-416  
 MEASUrement:IMMed:REFLevel:PERCent:  
   LOW, 2-417  
 MEASUrement:IMMed:REFLevel:PERCent:  
   MID<x>, 2-417  
 MEASUrement:IMMed:REFLevel?, 2-413  
 MEASUrement:IMMed:SOUrce<x>, 2-418  
 MEASUrement:IMMed:SOUrce<x>:SIGType, 2-419  
 MEASUrement:IMMed:TYPE, 2-419  
 MEASUrement:IMMed:UNIts?, 2-424  
 MEASUrement:IMMed:VALue?, 2-424  
 MEASUrement:MEAS<x>:COUNT?, 2-425  
 MEASUrement:MEAS<x>:DELay:DIREction, 2-426  
 MEASUrement:MEAS<x>:DELay:EDGE<x>, 2-427  
 MEASUrement:MEAS<x>:DELay?, 2-426  
 MEASUrement:MEAS<x>:MAXimum?, 2-427  
 MEASUrement:MEAS<x>:MEAN?, 2-428  
 MEASUrement:MEAS<x>:METHod, 2-428  
 MEASUrement:MEAS<x>:MINimum?, 2-429  
 MEASUrement:MEAS<x>:NOISe, 2-429  
 MEASUrement:MEAS<x>:REFLevel:ABSolute:  
   HIGH, 2-430  
 MEASUrement:MEAS<x>:REFLevel:ABSolute:  
   LOW, 2-431  
 MEASUrement:MEAS<x>:REFLevel:ABSolute:  
   MID<x>, 2-431  
 MEASUrement:MEAS<x>:REFLevel:  
   METHod, 2-432  
 MEASUrement:MEAS<x>:REFLevel:PERCent:  
   HIGH, 2-433  
 MEASUrement:MEAS<x>:REFLevel:PERCent:  
   LOW, 2-433  
 MEASUrement:MEAS<x>:REFLevel:PERCent:  
   MID<x>, 2-434  
 MEASUrement:MEAS<x>:REFLevel?, 2-430  
 MEASUrement:MEAS<x>:SOUrce<x>, 2-435  
 MEASUrement:MEAS<x>:SOUrce<x>:  
   SIGType, 2-435  
 MEASUrement:MEAS<x>:STATE, 2-436  
 MEASUrement:MEAS<x>:STDdev?, 2-437  
 MEASUrement:MEAS<x>:TYPE, 2-437  
 MEASUrement:MEAS<x>:UNIts?, 2-442  
 MEASUrement:MEAS<x>:VALue?, 2-442  
 MEASUrement:MEAS<x>?, 2-425  
 MEASUrement:REFLevel:ABSolute:HIGH, 2-444  
 MEASUrement:REFLevel:ABSolute:LOW, 2-445  
 MEASUrement:REFLevel:ABSolute:MID<x>, 2-446  
 MEASUrement:REFLevel:METHod, 2-446  
 MEASUrement:REFLevel:PERCent:HIGH, 2-447  
 MEASUrement:REFLevel:PERCent:LOW, 2-448  
 MEASUrement:REFLevel:PERCent:MID<x>, 2-449  
 MEASUrement:SOUrce<x>:SIGType, 2-450  
 MEASUrement:STATIstics:COUNT, 2-450  
 MEASUrement:STATIstics:MODE, 2-451  
 MEASUrement:STATIstics:WEIGHting, 2-451

MEASUREMENT?, 2-404  
 MEASUREMENT:GATING, 2-408  
 MEASUREMENT:IMMED?, 2-409  
 MEASUREMENT:METHOd, 2-443  
 MEASUREMENT:NOISE, 2-444

## N

NEWpass, 2-452

## O

\*OPC, 2-452  
 \*OPT?, 2-453

## P

PASSWord, 2-454  
 \*PSC, 2-454  
 \*PUD, 2-455

## R

\*RCL, 2-455  
 RECALL:MASK, 2-456  
 RECALL:SETUp, 2-457  
 RECALL:SETUp:DESKew, 2-457  
 RECALL:WAVEform, 2-458  
 REF<x>:HORizontal:POSition, 2-459  
 REF<x>:LAbel:NAME, 2-460  
 REF<x>:LAbel:XPOS, 2-460  
 REF<x>:LAbel:YPOS, 2-461  
 REF<x>:VERTical:POSition, 2-462  
 REF<x>:VERTical:SCALE, 2-463  
 REF<x>:LAbel?, 2-459  
 REF<x>:THRESHold, 2-462  
 REM, 2-464  
 ROSc:OUT:FREQUency, 2-464  
 ROSc:SOURce, 2-465  
 ROSc:STATE?, 2-465  
 ROSc:TRACKing, 2-466  
 \*RST, 2-467

## S

\*SAV, 2-467  
 SAVe:WAVEform:FILEFormat, 2-470  
 SAVe:MASK, 2-468  
 SAVe:SETUp, 2-468  
 SAVe:WAVEform, 2-469

\*SDS, 2-471  
 SEARCH:SEARCH<x>:COpy, 2-472  
 SEARCH:SEARCH<x>:STATE, 2-473  
 SEARCH:SEARCH<x>:TOTAL?, 2-473  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   AUTOBitrate, 2-474  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   AUTOReflevel, 2-474  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   BITRate, 2-475  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   CSActive, 2-475  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   CSLevel, 2-476  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   CSMode, 2-476  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   CSSource, 2-477  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   DATASource, 2-478  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   HYSterisis, 2-478  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   MARgin, 2-479  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   MAXCas, 2-479  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   MINCas, 2-480  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:DATA:HIGH, 2-480  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:DATA:LOW, 2-481  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:DATA:MID, 2-481  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:STROBE:HIGH, 2-482  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:STROBE:LOW, 2-482  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   REFLevel:STROBE:MID, 2-483  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   STANdard, 2-484  
 SEARCH:SEARCH<x>:TRIGger:A:DDRMemory:  
   STROBEsource, 2-484  
 SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:  
   CH<x>, 2-485

- SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:  
MATH<x>, 2-485
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe:  
REF<x>, 2-486
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:  
SOUrce, 2-487
- SEARCH:SEARCH<x>:TRIGger:A:GLItch:  
POLarity:CH<x>, 2-487
- SEARCH:SEARCH<x>:TRIGger:A:GLItch:  
POLarity:MATH<x>, 2-488
- SEARCH:SEARCH<x>:TRIGger:A:GLItch:  
POLarity:REF<x>, 2-488
- SEARCH:SEARCH<x>:TRIGger:A:GLItch:  
TRIGIF, 2-489
- SEARCH:SEARCH<x>:TRIGger:A:GLItch:  
WIDth, 2-489
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
CH<x>, 2-490
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
MATH<x>, 2-490
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
REF<x>, 2-491
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
FUNctIon, 2-492
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
CH<x>, 2-492
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
MATH<x>, 2-493
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
REF<x>, 2-493
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:CH<x>, 2-494
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:MATH<x>, 2-494
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:REF<x>, 2-495
- SEARCH:SEARCH<x>:TRIGger:A:PATtern:  
WHEn, 2-495
- SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:  
LESSLimit, 2-496
- SEARCH:SEARCH<x>:TRIGger:A:PATtern:WHEn:  
MORELimit, 2-497
- SEARCH:SEARCH<x>:TRIGger:A:PULse:  
SOUrce, 2-497
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
POLarity:CH<x>, 2-498
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
POLarity:MATH<x>, 2-498
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
POLarity:REF<x>, 2-499
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:HIGH:CH<x>, 2-500
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:HIGH:MATH<x>, 2-500
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:HIGH:REF<x>, 2-501
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:LOW:CH<x>, 2-501
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:LOW:MATH<x>, 2-502
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
THReshold:LOW:REF<x>, 2-502
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WHEn, 2-503
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WIDth, 2-503
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:EDGE, 2-504
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:SOUrce, 2-504
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:THReshold:CH<x>, 2-505
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:THReshold:MATH<x>, 2-506
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:THReshold:REF<x>, 2-506
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
SOUrce, 2-507
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
THReshold:CH<x>, 2-507
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
THReshold:MATH<x>, 2-508
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
THReshold:REF<x>, 2-509
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
HOLDTime, 2-509
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
SETTime, 2-510
- SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:  
EDGE, 2-510
- SEARCH:SEARCH<x>:TRIGger:A:STATE:CLOCK:  
SOUrce, 2-511

- SEARCH:SEARCH<x>:TRIGger:A:STATE:  
WHEn, 2-511
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:  
POLarity:CH<x>, 2-512
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:  
POLarity:MATH<x>, 2-512
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:  
POLarity:REF<x>, 2-513
- SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:  
TIME, 2-514
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
DELTATime, 2-514
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
POLarity:CH<x>, 2-515
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
POLarity:MATH<x>, 2-515
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
POLarity:REF<x>, 2-516
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:HIGH:CH<x>, 2-517
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:HIGH:MATH<x>, 2-517
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:HIGH:REF<x>, 2-518
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:LOW:CH<x>, 2-518
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:LOW:MATH<x>, 2-519
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
THReshold:LOW:REF<x>, 2-519
- SEARCH:SEARCH<x>:TRIGger:A:TRANSition:  
WHEn, 2-520
- SEARCH:SEARCH<x>:TRIGger:A:TYPE, 2-520
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
HIGHLimit, 2-522
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
LOWLimit, 2-522
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
POLarity:CH<x>, 2-523
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
POLarity:MATH<x>, 2-523
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
POLarity:REF<x>, 2-524
- SEARCH:SEARCH<x>:TRIGger:A:WIDTH:  
WHEn, 2-524
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
EVENT, 2-525
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:HIGH:CH<x>, 2-526
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:HIGH:MATH<x>, 2-526
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:HIGH:REF<x>, 2-527
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:LOW:CH<x>, 2-527
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:LOW:MATH<x>, 2-528
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
THReshold:LOW:REF<x>, 2-528
- SEARCH:SEARCH<x>:TRIGger:A:WINdow:  
WIDTH, 2-529
- SEARCH:SEARCH<x>, 2-472
- SEARCH:STOP, 2-529
- SElect:DIGTraces:COMBination, 2-532
- SElect:DIGTraces:LIST, 2-533
- SElect?, 2-530
- SElect:<wfm>, 2-534
- SElect:B<x>, 2-530
- SElect:CH<x>, 2-531
- SElect:CONTROL, 2-531
- SElect:D<x>, 2-533
- SElect:DALL, 2-532
- SET?, 2-534
- SETUp:NAME, 2-535
- \*SRE, 2-536
- \*STB?, 2-537
- ## T
- TEKLink:CONNection?, 2-537
- TEKLink:REFClk, 2-537
- TEKSecure, 2-538
- TEST, 2-538
- TEST:RESults:VERBose?, 2-540
- TEST:RESults?, 2-539
- TEST:STOP, 2-540
- TIME, 2-540
- \*TRG, 2-541
- TRIGger, 2-541
- TRIGger: {A|B}:EDGE:COUPling, 2-543
- TRIGger: {A|B}:EDGE:COUPling:CH<x>, 2-544
- TRIGger: {A|B}:EDGE:SLOpe, 2-545
- TRIGger: {A|B}:EDGE:SLOpe:AUX, 2-545
- TRIGger: {A|B}:EDGE:SLOpe:CH<x>, 2-546
- TRIGger: {A|B}:EDGE:SOUrce, 2-546

- TRIGger: {A|B}:LEVel:CH<x>, 2-548  
 TRIGger: {A|B}:LOGIc:CLAss, 2-549  
 TRIGger: {A|B}:LOGIc:FUNcTION, 2-550  
 TRIGger: {A|B}:LOGIc:INPut:ALL, 2-551  
 TRIGger: {A|B}:LOGIc:INPut:CH<x>, 2-551  
 TRIGger: {A|B}:LOGIc:INPut:FORMAt, 2-552  
 TRIGger: {A|B}:LOGIc:INPut?, 2-550  
 TRIGger: {A|B}:LOGIc:PATtern:INPut:CH<x>, 2-553  
 TRIGger: {A|B}:LOGIc:PATtern:WHEN, 2-554  
 TRIGger: {A|B}:LOGIc:PATtern:WHEN:  
   LESSLimit, 2-555  
 TRIGger: {A|B}:LOGIc:PATtern:WHEN:  
   MORELimit, 2-555  
 TRIGger: {A|B}:LOGIc:PATtern?, 2-553  
 TRIGger: {A|B}:LOGIc:SETHold:CLOCK:  
   EDGE, 2-557  
 TRIGger: {A|B}:LOGIc:SETHold:CLOCK:  
   SOURCE, 2-558  
 TRIGger: {A|B}:LOGIc:SETHold:CLOCK:  
   THReshold, 2-558  
 TRIGger: {A|B}:LOGIc:SETHold:CLOCK:  
   THReshold:CH<x>, 2-559  
 TRIGger: {A|B}:LOGIc:SETHold:CLOCK?, 2-557  
 TRIGger: {A|B}:LOGIc:SETHold:DATA:  
   SOURCE, 2-560  
 TRIGger: {A|B}:LOGIc:SETHold:DATA:  
   THReshold, 2-561  
 TRIGger: {A|B}:LOGIc:SETHold:DATA:THReshold:  
   CH<x>, 2-561  
 TRIGger: {A|B}:LOGIc:SETHold:DATA?, 2-560  
 TRIGger: {A|B}:LOGIc:SETHold:HOLDTime, 2-562  
 TRIGger: {A|B}:LOGIc:SETHold:QUALify, 2-562  
 TRIGger: {A|B}:LOGIc:SETHold:SETTime, 2-563  
 TRIGger: {A|B}:LOGIc:SETHold?, 2-556  
 TRIGger: {A|B}:LOGIc:STATE:CLOCK:  
   SOURCE, 2-564  
 TRIGger: {A|B}:LOGIc:STATE:INPut:CH<x>, 2-565  
 TRIGger: {A|B}:LOGIc:STATE:WHEN, 2-565  
 TRIGger: {A|B}:LOGIc:STATE?, 2-564  
 TRIGger: {A|B}:LOGIc:THReshold:CH<x>, 2-567  
 TRIGger: {A|B}:LOGIc:THReshold?, 2-566  
 TRIGger: {A|B}:LOGIc?, 2-548  
 TRIGger: {A|B}:LOWerthreshold:CH<x>, 2-567  
 TRIGger: {A|B}:PULse:CLAss, 2-568  
 TRIGger: {A|B}:PULse:GLItch:POLarity, 2-570  
 TRIGger: {A|B}:PULse:GLItch:POLarity:  
   CH<x>, 2-571  
 TRIGger: {A|B}:PULse:GLItch:QUALify, 2-571  
 TRIGger: {A|B}:PULse:GLItch:TRIGIF, 2-572  
 TRIGger: {A|B}:PULse:GLItch:WIDth, 2-573  
 TRIGger: {A|B}:PULse:GLItch?, 2-569  
 TRIGger: {A|B}:PULse:RUNT:POLarity, 2-574  
 TRIGger: {A|B}:PULse:RUNT:POLarity:  
   CH<x>, 2-575  
 TRIGger: {A|B}:PULse:RUNT:QUALify, 2-575  
 TRIGger: {A|B}:PULse:RUNT:THReshold:  
   BOTH, 2-577  
 TRIGger: {A|B}:PULse:RUNT:THReshold:  
   HIGH, 2-577  
 TRIGger: {A|B}:PULse:RUNT:THReshold:  
   LOW, 2-578  
 TRIGger: {A|B}:PULse:RUNT:THReshold?, 2-576  
 TRIGger: {A|B}:PULse:RUNT:WHEN, 2-578  
 TRIGger: {A|B}:PULse:RUNT:WIDth, 2-579  
 TRIGger: {A|B}:PULse:RUNT?, 2-573  
 TRIGger: {A|B}:PULse:SOURCE, 2-579  
 TRIGger: {A|B}:PULse:TIMEOut:POLarity, 2-580  
 TRIGger: {A|B}:PULse:TIMEOut:POLarity:  
   CH<x>, 2-581  
 TRIGger: {A|B}:PULse:TIMEOut:QUALify, 2-582  
 TRIGger: {A|B}:PULse:TIMEOut:TIME, 2-583  
 TRIGger: {A|B}:PULse:TIMEOut?, 2-580  
 TRIGger: {A|B}:PULse:TRANSition:  
   DELTAime, 2-584  
 TRIGger: {A|B}:PULse:TRANSition:POLarity, 2-584  
 TRIGger: {A|B}:PULse:TRANSition:POLarity:  
   CH<x>, 2-585  
 TRIGger: {A|B}:PULse:TRANSition:QUALify, 2-586  
 TRIGger: {A|B}:PULse:TRANSition:THReshold:  
   BOTH, 2-587  
 TRIGger: {A|B}:PULse:TRANSition:THReshold:  
   HIGH, 2-587  
 TRIGger: {A|B}:PULse:TRANSition:THReshold:  
   LOW, 2-588  
 TRIGger: {A|B}:PULse:TRANSition:  
   THReshold?, 2-586  
 TRIGger: {A|B}:PULse:TRANSition:WHEN, 2-588  
 TRIGger: {A|B}:PULse:TRANSition?, 2-583  
 TRIGger: {A|B}:PULse:WIDth:HIGHLimit, 2-590  
 TRIGger: {A|B}:PULse:WIDth:LOWLimit, 2-590  
 TRIGger: {A|B}:PULse:WIDth:POLarity, 2-591  
 TRIGger: {A|B}:PULse:WIDth:POLarity:  
   CH<x>, 2-591  
 TRIGger: {A|B}:PULse:WIDth:QUALify, 2-592

- TRIGger: {A|B}:PULse:WIDth:WHEn, 2-593  
 TRIGger: {A|B}:PULse:WIDth?, 2-589  
 TRIGger: {A|B}:PULse:WINDow:EVENT, 2-594  
 TRIGger: {A|B}:PULse:WINDow:POLarity, 2-595  
 TRIGger: {A|B}:PULse:WINDow:POLarity:  
   CH<x>, 2-595  
 TRIGger: {A|B}:PULse:WINDow:QUALify, 2-596  
 TRIGger: {A|B}:PULse:WINDow:THReshold:  
   BOTH, 2-597  
 TRIGger: {A|B}:PULse:WINDow:THReshold:  
   HIGH, 2-598  
 TRIGger: {A|B}:PULse:WINDow:THReshold:  
   LOW, 2-598  
 TRIGger: {A|B}:PULse:WINDow:THReshold?, 2-597  
 TRIGger: {A|B}:PULse:WINDow:WIDTh, 2-599  
 TRIGger: {A|B}:PULse:WINDow?, 2-594  
 TRIGger: {A|B}:PULse?, 2-568  
 TRIGger: {A|B}:READY?, 2-599  
 TRIGger: {A|B}:UPPerthreshold:CH<x>, 2-601  
 TRIGger:A:BUS:DATA:FORMAt, 2-603  
 TRIGger:A:BUS:DATA:VALue, 2-604  
 TRIGger:A:BUS:I2C:ADDRes:FORMAt, 2-604  
 TRIGger:A:BUS:I2C:ADDRes:MODE, 2-605  
 TRIGger:A:BUS:I2C:ADDRes:VALue, 2-605  
 TRIGger:A:BUS:I2C:CONDition, 2-606  
 TRIGger:A:BUS:I2C:DATA:DIRection, 2-607  
 TRIGger:A:BUS:SOURce, 2-607  
 TRIGger:A:BUS:SPI:CONDition, 2-608  
 TRIGger:A:CAN:CONDition, 2-608  
 TRIGger:A:CAN:DATA:DIRection, 2-609  
 TRIGger:A:CAN:DATA:LEVel, 2-610  
 TRIGger:A:CAN:DATA:SOURce, 2-610  
 TRIGger:A:CAN:DATA:VALue, 2-610  
 TRIGger:A:CAN:FORMAt, 2-611  
 TRIGger:A:CAN:FRAMetype, 2-611  
 TRIGger:A:CAN:IDENTifier:MODE, 2-612  
 TRIGger:A:CAN:IDENTifier:VALue, 2-612  
 TRIGger:A:COMMunication:  
   {AMI|HDB3|B3ZS|B6ZS|B8ZS}:  
   PULSEForm, 2-614  
 TRIGger:A:COMMunication:  
   {AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:  
   HIGH, 2-614  
 TRIGger:A:COMMunication:  
   {AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:  
   LOW, 2-615  
 TRIGger:A:COMMunication:BITRate, 2-616  
 TRIGger:A:COMMunication:CLOCK:  
   POLarity, 2-616  
 TRIGger:A:COMMunication:CMI:  
   PULSEForm, 2-617  
 TRIGger:A:COMMunication:CODE, 2-617  
 TRIGger:A:COMMunication:SOURce, 2-618  
 TRIGger:A:COMMunication:SOURce:TYPE, 2-619  
 TRIGger:A:COMMunication:STANdard, 2-619  
 TRIGger:A:HOLDoff:ACTUal?, 2-623  
 TRIGger:A:HOLDoff:BY, 2-623  
 TRIGger:A:HOLDoff:TIME, 2-624  
 TRIGger:A:I2C:ADDRes:MODE, 2-624  
 TRIGger:A:I2C:ADDRes:RWINclude, 2-625  
 TRIGger:A:I2C:ADDRes:TYPE, 2-625  
 TRIGger:A:I2C:ADDRes:VALue, 2-626  
 TRIGger:A:I2C:CLOCK:LEVel, 2-626  
 TRIGger:A:I2C:CLOCK:SOURce, 2-627  
 TRIGger:A:I2C:CONDition, 2-627  
 TRIGger:A:I2C:DATA:DIRection, 2-628  
 TRIGger:A:I2C:DATA:LEVel, 2-628  
 TRIGger:A:I2C:DATA:SOURce, 2-629  
 TRIGger:A:I2C:DATA:STARt, 2-629  
 TRIGger:A:I2C:DATA:VALue, 2-630  
 TRIGger:A:I2C:FORMAt, 2-630  
 TRIGger:A:RS232:BAUd, 2-631  
 TRIGger:A:RS232:DATA:LEVel, 2-632  
 TRIGger:A:RS232:DATA:SOURce, 2-632  
 TRIGger:A:RS232:DATA:VALue, 2-633  
 TRIGger:A:RS232:FORMAt, 2-633  
 TRIGger:A:RS232:PARity, 2-634  
 TRIGger:A:SERIAL:BITRate, 2-634  
 TRIGger:A:SERIAL:CLOCK:LEVel, 2-635  
 TRIGger:A:SERIAL:CLOCK:POLarity, 2-636  
 TRIGger:A:SERIAL:CLOCK:SOURce, 2-637  
 TRIGger:A:SERIAL:CODE, 2-637  
 TRIGger:A:SERIAL:DATA:FORMAt, 2-638  
 TRIGger:A:SERIAL:DATA:PATtern, 2-638  
 TRIGger:A:SERIAL:DATA:PATtern:NRZ, 2-639  
 TRIGger:A:SERIAL:DATA:PATtern:S8B10B, 2-640  
 TRIGger:A:SERIAL:ERRORdetector:  
   ALIGNCHARacter:RDMINUS, 2-642  
 TRIGger:A:SERIAL:ERRORdetector:  
   ALIGNCHARacter:RDPLUS, 2-642  
 TRIGger:A:SERIAL:ERRORdetector:  
   ALIGNCHARacter?, 2-641  
 TRIGger:A:SERIAL:ERRORdetector:  
   ALIGNPRIMitive:RDMINUS:ENABLE, 2-643



- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:RDMINUS:PATtern, 2-644
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:RDMINUS?, 2-643
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:RDPLUS:ENABLE, 2-645
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:RDPLUS:PATtern, 2-645
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:RDPLUS?, 2-644
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive:S8BPATtern, 2-646
- TRIGger:A:SERIAL:ERRORdetector:  
ALIGNPRIMitive?, 2-643
- TRIGger:A:SERIAL:ERRORdetector:BIT:  
LENGTH, 2-647
- TRIGger:A:SERIAL:ERRORdetector:BIT:  
SYNCPATtern, 2-647
- TRIGger:A:SERIAL:ERRORdetector:BIT:  
TEST, 2-648
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
COUNT?, 2-648
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
DURATION?, 2-648
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
MAXALIGNs?, 2-649
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
RATE?, 2-649
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
RESULTS?, 2-649
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
SECONDS?, 2-650
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS:LOCK?, 2-651
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS:MAX\_AP?, 2-651
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS:SIGNAL?, 2-652
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS:START?, 2-652
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS:SYNC?, 2-652
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
STATUS?, 2-650
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
TIME:DAYs?, 2-653
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
TIME:HOURS?, 2-653
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
TIME:MINUTES?, 2-654
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
TIME:SECONDS?, 2-654
- TRIGger:A:SERIAL:ERRORdetector:BIT:TEST:  
TIME?, 2-653
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST, 2-655
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:COUNT?, 2-655
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:DISPCOUNT?, 2-656
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:DISPRATE?, 2-656
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:DURATION?, 2-656
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:MAXALIGNs?, 2-657
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:RATE?, 2-657
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:RESULTS?, 2-658
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:SECONDS?, 2-658
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:STATUS:LOCK?, 2-660
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:STATUS:MAX\_AP?, 2-659
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:STATUS:SIGNAL?, 2-660
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:STATUS:START?, 2-660
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:STATUS?, 2-659
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:TIME:DAYs?, 2-661
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:TIME:HOURS?, 2-662
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:TIME:MINUTES?, 2-662
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:TIME:SECONDS?, 2-662
- TRIGger:A:SERIAL:ERRORdetector:CHARACTER:  
TEST:TIME?, 2-661

- TRIGger:A:SERIAL:ERRORdetector:  
CHARACTER?, 2-654
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
COUNT, 2-663
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
SECONDS, 2-663
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
TIME, 2-664
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
TIME:DAYS, 2-664
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
TIME:HOURS, 2-665
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
TIME:MINUTES, 2-665
- TRIGger:A:SERIAL:ERRORdetector:DURATION:  
TIME:SECONDS, 2-666
- TRIGger:A:SERIAL:ERRORdetector:ENable, 2-666
- TRIGger:A:SERIAL:ERRORdetector:  
ERRORLIMIT, 2-667
- TRIGger:A:SERIAL:ERRORdetector:FILE:  
RECALL, 2-667
- TRIGger:A:SERIAL:ERRORdetector:FILE:  
SAVE, 2-668
- TRIGger:A:SERIAL:ERRORdetector:FRAME:  
EOF, 2-669
- TRIGger:A:SERIAL:ERRORdetector:FRAME:  
INITIALCRCVALue, 2-669
- TRIGger:A:SERIAL:ERRORdetector:FRAME:  
SOF, 2-670
- TRIGger:A:SERIAL:ERRORdetector:FRAME:  
TEST, 2-670
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
BADCHARS?, 2-671
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
COUNT?, 2-671
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
DISPARITY?, 2-671
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
DURATION?, 2-672
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
MAXALIGNs?, 2-672
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
RATE?, 2-672
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
RESULTS?, 2-673
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
SECONDS?, 2-673
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
STATUS:LOCK?, 2-675
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
STATUS:MAX\_AP?, 2-674
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
STATUS:SIGNAL?, 2-675
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
STATUS:START?, 2-675
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
STATUS?, 2-674
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
TIME:DAYS?, 2-676
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
TIME:HOURS?, 2-676
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
TIME:MINUTES?, 2-677
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
TIME:SECONDS?, 2-677
- TRIGger:A:SERIAL:ERRORdetector:FRAME:TEST:  
TIME?, 2-676
- TRIGger:A:SERIAL:ERRORdetector:  
FRAME?, 2-668
- TRIGger:A:SERIAL:ERRORdetector:  
MAXALIGNs, 2-677
- TRIGger:A:SERIAL:ERRORdetector:  
PATTERNNAME, 2-678
- TRIGger:A:SERIAL:ERRORdetector:  
SCRAMBLED, 2-678
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDMINUS:ENABLE, 2-679
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDMINUS:PATtern, 2-680
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDMINUS?, 2-679
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDPLUS:ENABLE, 2-681
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDPLUS:PATtern, 2-681
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:RDPLUS?, 2-680
- TRIGger:A:SERIAL:ERRORdetector:  
SKIPSETPRIMitive:S8BPATtern, 2-682
- TRIGger:A:SERIAL:ERRORdetector:SSC, 2-682
- TRIGger:A:SERIAL:ERRORdetector:  
STANDARD, 2-683
- TRIGger:A:SERIAL:ERRORdetector:  
STOPWHEN, 2-683

- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST, 2-684
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:BADCHARS?, 2-685
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:BITCOUNT?, 2-685
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:BITDURATION?, 2-686
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:BITRATE?, 2-686
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:COUNT?, 2-686
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:DISPARITY?, 2-687
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:DURATION?, 2-687
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:MAXALIGNs?, 2-687
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:RATE?, 2-688
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:RESULTS?, 2-688
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:SECONDS?, 2-689
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:STATUS:LOCK?, 2-690
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:STATUS:MAX\_AP?, 2-690
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:STATUS:SIGNAL?, 2-691
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:STATUS:START?, 2-691
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:STATUS?, 2-689
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:TIME:DAYS?, 2-692
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:TIME:HOURS?, 2-692
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:TIME:MINUTES?, 2-692
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:TIME:SECONDS?, 2-693
- TRIGger:A:SERIAL:ERRORdetector:SYMBOL:  
TEST:TIME?, 2-691
- TRIGger:A:SERIAL:ERRORdetector:  
SYMBOL?, 2-684
- TRIGger:A:SERIAL:ERRORdetector:TYPE, 2-693
- TRIGger:A:SERIAL:LOCKLen, 2-694
- TRIGger:A:SERIAL:LOCKOffset, 2-694
- TRIGger:A:SERIAL:SOUrce, 2-695
- TRIGger:A:SERIAL:STANdard, 2-695
- TRIGger:A:SERIAL:TRIGgeron, 2-697
- TRIGger:A:SPI:CONDition, 2-697
- TRIGger:A:SPI:DATA:MISO:ACTIVE, 2-698
- TRIGger:A:SPI:DATA:MISO:LEVel, 2-698
- TRIGger:A:SPI:DATA:MISO:SOUrce, 2-699
- TRIGger:A:SPI:DATA:MISO:VALue, 2-699
- TRIGger:A:SPI:DATA:MOSI:ACTIVE, 2-700
- TRIGger:A:SPI:DATA:MOSI:LEVel, 2-700
- TRIGger:A:SPI:DATA:MOSI:SOUrce, 2-701
- TRIGger:A:SPI:DATA:MOSI:VALue, 2-701
- TRIGger:A:SPI:DATA:STARt, 2-702
- TRIGger:A:SPI:FORMAt, 2-702
- TRIGger:A:SPI:SCLK:ACTIVE, 2-703
- TRIGger:A:SPI:SCLK:LEVel, 2-703
- TRIGger:A:SPI:SCLK:SOUrce, 2-704
- TRIGger:A:SPI:SS:ACTIVE, 2-704
- TRIGger:A:SPI:SS:LEVel, 2-705
- TRIGger:A:SPI:SS:SOUrce, 2-705
- TRIGger:A:VIDeo:CUSTom:FORMAt, 2-706
- TRIGger:A:VIDeo:CUSTom:LINEPeriod, 2-707
- TRIGger:A:VIDeo:CUSTom:SYNCInterval, 2-707
- TRIGger:A:VIDeo:CUSTom?, 2-706
- TRIGger:A:VIDeo:FIELD, 2-708
- TRIGger:A:VIDeo:HOLdoff:FIELD, 2-708
- TRIGger:A:VIDeo:LINE, 2-709
- TRIGger:A:VIDeo:POLarity, 2-710
- TRIGger:A:VIDeo:SCAN, 2-711
- TRIGger:A:VIDeo:SOUrce, 2-711
- TRIGger:A:VIDeo:STANdard, 2-712
- TRIGger:B:EVENTS:COUNt, 2-716
- TRIGger:B:RESET:SOUrce, 2-716
- TRIGger:B:RESET:STATE, 2-717
- TRIGger:B:RESET:THReshold, 2-717
- TRIGger:B:RESET:TIMEOut, 2-718
- TRIGger:B:RESET:TRANSition, 2-719
- TRIGger:B:RESET:TYPe, 2-719
- TRIGger:B:SCAN:ADVANCEafter, 2-720
- TRIGger:B:SCAN:ENABle, 2-720
- TRIGger:B:SCAN:ENDEvent, 2-721
- TRIGger:B:SCAN:STARTEvent, 2-722
- TRIGger:LVLsrcpreference, 2-724
- TRIGger:MULTiscope:ALIGN, 2-726
- TRIGger:MULTiscope:ALIGN:COMPLeted?, 2-726

TRIGger:MULTiscope:ALIGN:DESKEW?, 2-726  
TRIGger:MULTiscope:ALIGN:MRTTime, 2-727  
TRIGger:MULTiscope:ALIGN:SETSTATE, 2-727  
TRIGger:MULTiscope:ALIGN:VALue?, 2-728  
TRIGger:MULTiscope:DELay?, 2-728  
TRIGger:MULTiscope:LOGic, 2-728  
TRIGger:MULTiscope:ROLe, 2-729  
TRIGger:QUALification:BUS:FORMat, 2-730  
TRIGger:QUALification:BUS:SOURce, 2-730  
TRIGger:QUALification:BUS:VALue, 2-731  
TRIGger:{A|B}:EDGE?, 2-543  
TRIGger:{A|B}:LEVel, 2-547  
TRIGger:{A|B}:TYPe, 2-600  
TRIGger:A, 2-602  
TRIGger:A:CAN:PROBE, 2-613  
TRIGger:A:CAN:SPEEd, 2-613  
TRIGger:A:HOLDoff?, 2-622  
TRIGger:A:MODE, 2-631  
TRIGger:A:VIDeo?, 2-705  
TRIGger:AUXLevel, 2-713  
TRIGger:B, 2-714  
TRIGger:B:BY, 2-714  
TRIGger:B:EVENTS?, 2-715  
TRIGger:B:SCAN:MODE, 2-722  
TRIGger:B:STATE, 2-723  
TRIGger:B:TIME, 2-723  
TRIGger:ENHanced, 2-724  
TRIGger:MULTiscope, 2-725  
TRIGger:SENSITivity, 2-731  
TRIGger:STATE?, 2-732  
\*TST?, 2-733

## U

UNLock, 2-733

## V

VERBose, 2-734

## W

\*WAI, 2-734  
WAVFrm?, 2-735  
WAVFRMStream?, 2-736  
WFMInpre?, 2-737  
WFMInpre:BIT\_Nr, 2-738  
WFMInpre:BN\_Fmt, 2-738  
WFMInpre:BYT\_Nr, 2-739

WFMInpre:BYT\_Or, 2-739  
WFMInpre:ENCdg, 2-740  
WFMInpre:NR\_FR, 2-740  
WFMInpre:NR\_Pt, 2-741  
WFMInpre:PT\_Fmt, 2-741  
WFMInpre:PT\_Off, 2-742  
WFMInpre:WFId, 2-743  
WFMInpre:XINcr, 2-743  
WFMInpre:XUNit, 2-744  
WFMInpre:XZEro, 2-744  
WFMInpre:YMUlt, 2-745  
WFMInpre:YOFF, 2-745  
WFMInpre:YUNit, 2-746  
WFMInpre:YZEro, 2-747  
WFMOutpre?, 2-747  
WFMOutpre:BIT\_Nr, 2-748  
WFMOutpre:BN\_Fmt, 2-748  
WFMOutpre:BYT\_Nr, 2-749  
WFMOutpre:BYT\_Or, 2-749  
WFMOutpre:ENCdg, 2-750  
WFMOutpre:NR\_FR?, 2-751  
WFMOutpre:NR\_Pt?, 2-751  
WFMOutpre:PT\_Fmt?, 2-751  
WFMOutpre:PT\_Off?, 2-752  
WFMOutpre:PT\_ORder?, 2-752  
WFMOutpre:WFId?, 2-753  
WFMOutpre:XINcr?, 2-754  
WFMOutpre:XUNit?, 2-755  
WFMOutpre:XZEro?, 2-755  
WFMOutpre:YMUlt?, 2-756  
WFMOutpre:YOFF?, 2-756  
WFMOutpre:YUNit?, 2-757  
WFMOutpre:YZEro?, 2-757

## Z

ZOOm, 2-757  
ZOOm:HORizontal:POSition, 2-759  
ZOOm:HORizontal:SCALE, 2-759  
ZOOm:MATH<x>:HORizontal:POSition, 2-759  
ZOOm:MATH<x>:HORizontal:SCALE, 2-760  
ZOOm:MATH<x>:VERTical:POSition, 2-760  
ZOOm:MATH<x>:VERTical:SCALE, 2-761  
ZOOm:REF<x>:HORizontal:POSition, 2-762  
ZOOm:REF<x>:HORizontal:SCALE, 2-762  
ZOOm:REF<x>:VERTical:POSition, 2-763  
ZOOm:REF<x>:VERTical:SCALE, 2-763  
ZOOm:SCROLL:DIREction, 2-764

ZOom:VERTical:POSition, 2-766  
ZOom:ZOOM<x>:<wfm>:DISplay, 2-769  
ZOom:ZOOM<x>:<wfm>:HORizontal:  
    POSition, 2-770  
ZOom:ZOOM<x>:<wfm>:HORizontal:SCAle, 2-770  
ZOom:ZOOM<x>:<wfm>:VERTical:  
    POSition, 2-771  
ZOom:ZOOM<x>:<wfm>:VERTical:SCAle, 2-771  
ZOom:ZOOM<x>:DCHAN:DISplay, 2-767  
ZOom:ZOOM<x>:SCROLLLock, 2-768  
ZOom:ZOOM<x>:STATE, 2-769  
ZOom:GRAticule:SIZE, 2-758  
ZOom:MODE, 2-761  
ZOom:SCROLL:LOCK, 2-764  
ZOom:SCROLL:SPEED, 2-765  
ZOom:STATE, 2-765  
ZOom:VERTical:SCAle, 2-766  
ZOom:ZOOM<x>, 2-767