DPO7000, DPO70000 and DSA70000 Series Digital Oscilloscopes Programmer Manual

077-0010-01

DPO7000, DPO70000 and DSA70000 Series
Digital Oscilloscopes
Programmer Manual

www.tektronix.com

077-0010-01

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:

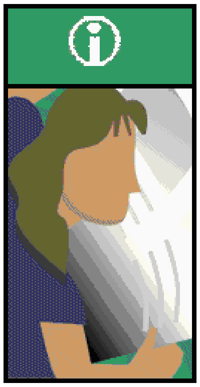
- In North America, call 1-800-833-9200.
- Worldwide, visit www.tektronix.com to find contacts in your area.

Table of Contents

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 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 module levels. This manual is available as a printable PDF file on the Product Software CD-ROM.

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.

Besides 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 assume that both `HEADER` and `VERBOSE` are `ON`.

Refer to Documentation for information on related manuals and documents.

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.

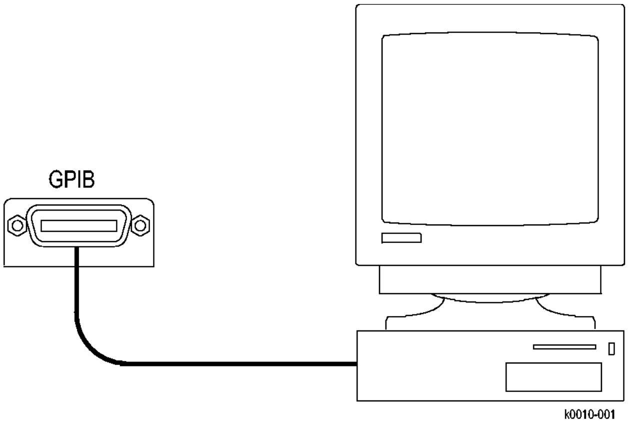
Setting Up Remote Communications

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

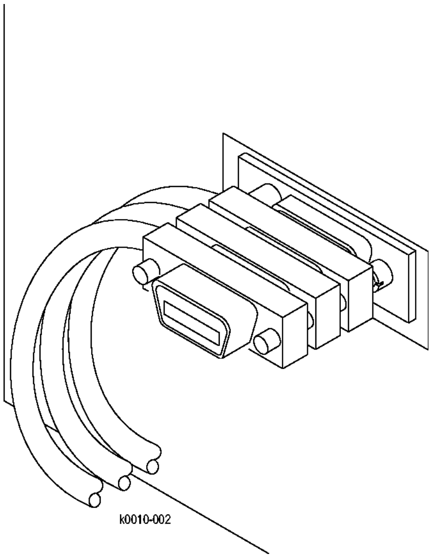
- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No more than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No more than 65 feet (20 meters) of cable should 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 488.1¾1987. Attach an IEEE Std 488.1¾1987 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 figure below.



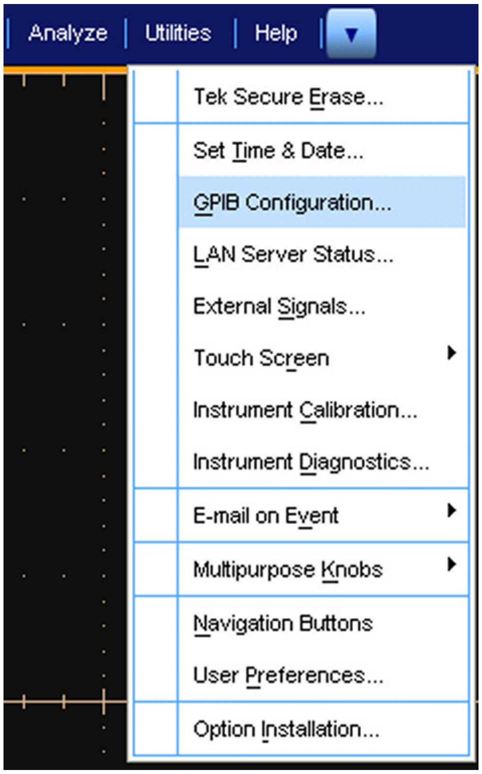
Setting the GPIB Address

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

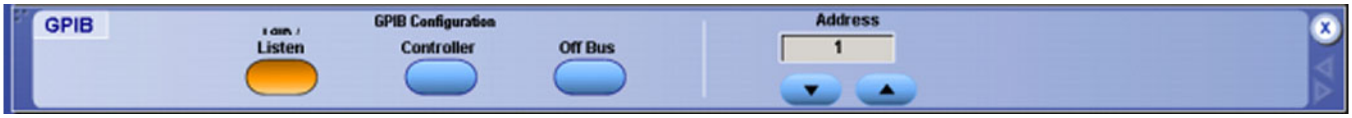
- GPIB Address 1
- GPIB ModeGPIB Talk/Listen

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 related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

Backus-Naur Form Notation

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

Table 2-1: Symbols for Backus-Naur Form

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

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

Commands

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

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

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

Queries

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

- `[:]<Header>?`
- `[:]<Header>?[<Space><Argument> [<Coma><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.

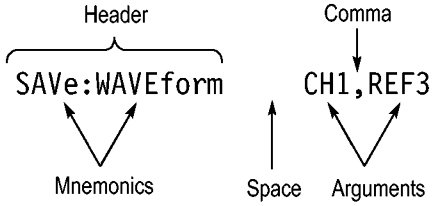
Headers

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 table below shows the difference in responses.

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

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

Command Parts



Clearing the Instrument

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 ACQuire:NUMAvg simply as ACQ:NUMA or acq:numa.

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

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

Concatenating

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

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

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:MODe NORMal` and `ACQuire:NUMAVg 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:STYle:NORMal;ACQuire:NUMAVg 10` (no colon before ACQuire)

`DISPlay:COLor:CURSor1 1;::CURSor2 5` (extra colon before `CURSor2;` use `DISPlay:COLor:CURSor1 1:CURSor2 5` instead)

`DISPlay:STYle:NORMal;:*OPC` (colon before a star (*) command)

`DISPlay:COLor:CURSor1 1;COLor:CURSor2 5` (levels of the mnemonics are different; either remove the second use of `COLor` or place `:DISPlay:` in front of `COLor:CURSor2 5`)

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

Table 2-10: Numeric Arguments, (cont.)

Symbol	Meaning
<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 as long as you follow the previous rule. For example, `"this is an 'acceptable' string"`.
3. You can include a quote character within a string by repeating the quote. For example: `"here is a "" mark"`.
4. Strings can have upper or lower case characters.
5. 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)

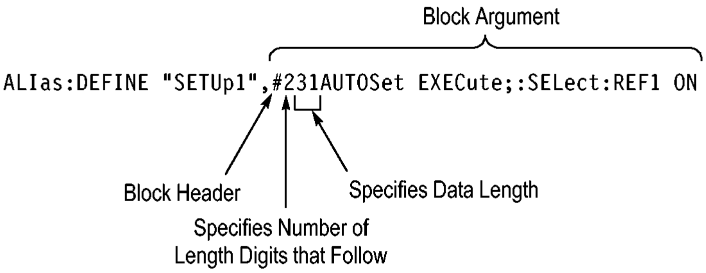
Block

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
<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.



```
ALIas:DEFINE "SETUp1",#231AUTOSet EXECute;:SELect:REF1 ON
```

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
ACQuire?	Returns acquisition parameters
ACQuire:MODe	Sets or returns acquisition mode
ACQuire:NUMACq?	Returns the number of acquisitions that have occurred
ACQuire:NUMAVg	Sets or returns number of acquisitions for an averaged waveform
ACQuire:NUMEnv	Sets or returns number of acquisitions for envelope waveform
ACQuire:NUMSAMples	Sets or returns the number of samples that make up a WfmDB for single sequence mode and Mask Pass/Fail Completion Test
ACQuire:SAMPlingmode	This command sets or queries the sampling mode
ACQuire:STATE	Starts, stops, or returns acquisition state
ACQuire:STOPAfter	Sets or returns whether the acquisition is continuous or single sequence

Table 2-13: Acquisition Commands, (cont.)

Command	Description
FASTAcq?	Enables, disables, or returns state of Fast Acquisition mode
FASTAcq:STATE	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
ALIas	Sets or returns the alias state
ALIas:CATalog?	Returns a list of the currently defined alias labels
ALIas:DEFine	Assigns a sequence of program messages to an alias label
ALIas:DELEte	Removes a specified alias
ALIas:DELEte:ALL	Deletes all existing aliases

Table 2-14: Alias Commands, (cont.)

Command	Description
ALIas:DELEte:NAMe	Removes a specified alias
ALIas:STATE	Sets or returns the alias state

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).

Command	Description
CALibrate?	Returns the internal and factory calibration status
*CAL?	Instructs the instrument to perform self-calibration and returns the calibration status when complete
CALibrate:CALProbe:CH<x>?	Performs a probe calibration for the selected channel and returns the calibration status
CALibrate:INTERNal	Starts the internal signal path calibration
CALibrate:INTERNal:STARt	Starts the internal signal path calibration
CALibrate:INTERNal:STATus?	Returns the current status of the internal signal path calibration
CALibrate:PRObestate:CH<x>?	Returns the probe calibration status for the probe of the selected channel
CALibrate:RESults?	Returns the status of all calibration subsystems without performing an SPC operation
CALibrate:RESults:SPC?	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 too.

Command	Description
CURSor?	Returns all cursor settings
CURSor:FUNCtion	Sets or returns the cursor type
CURSor:HBArs?	Returns hbar cursor settings
CURSor:HBArs:DELTa?	Returns hbars cursors vertical difference
CURSor:HBArs:POSITION<x>	Sets or returns the hbar cursor<x> vertical position
CURSor:HBArs:UNIts?	Returns hbar cursor units
CURSor:LINESTyle	Sets or returns the cursor line style
CURSor:MODe	Sets or returns whether cursors move in unison or separately
CURSor:SCREEN:STYle	Sets or returns 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 returns the source for cursor <x>
CURSor:STATE	Turns cursors on or off or returns their state
CURSor:VBArs	Sets or returns the position of vertical bar cursors

Command	Description
CURSor:VBArs:DELTa?	Returns the difference between vbar cursors
CURSor:VBArs:POSITION<x>	Sets or returns 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 returns the units for vbar cursors
CURSor:WAVEform	Sets or returns 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 returns the position of waveform cursor <x>
CURSor:WAVEform:SOUrce<x>	Sets or returns the source for waveform cursor <x>
CURSor:WAVEform:STYle	Sets or returns the cursor type for waveform mode
CURSor:WAVEform:UNIts	Sets or returns 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:PRODDELta?	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 returns the XY cursor readout mode
CURSor:XY:RECTX<x>	Sets or returns the X cursor position in rectangular coordinates
CURSor:XY:RECTY<x>	Sets or returns the Y cursor position in rectangular coordinates
CURSor:XY:THDELta?	Returns the XY cursor angle delta in polar coordinates
CURSor:XY:THETA<x>?	Returns the XY cursor angle in polar coordinates

Command	Description
CURSor:XY:XDELta?	Returns the XY cursor ΔX value in rectangular coordinates
CURSor:XY:YDELta?	Returns the XY cursor ΔY value in rectangular coordinates

Diagnostics Command Group

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

Command	Description
DIAg:CONTROL:HALT	Enables or disables halting on first diagnostic failure
DIAg:CONTROL:LOOP	Enables or disables looping of diagnostics
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
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

Command	Description
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

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 that the mode you choose globally affects all displayed waveforms.

Command	Description
DISplay?	Returns current display settings
DISplay:CLOCk	Sets or returns the display of the date/time stamp
DISplay:COLOr?	Returns color group settings
DISplay:COLOr:MATHCOLOr	Sets or returns the color to be used for math traces
DISplay:COLOr:PALEtte:IMAGEView	Sets or returns the color palette for imageview waveforms
DISplay:COLOr:PALEtte:RECORDView	Sets or returns the color palette for recordview waveforms
DISplay:COLOr:PALEtte:USEr	Returns the user palette group settings
DISplay:COLOr:PALEtte:USEr:CARet	Sets or returns the user caret color
DISplay:COLOr:PALEtte:USEr:CH<x>	Sets or returns the user palette channel colors
DISplay:COLOr:PALEtte:USEr:GRAticule	Sets or returns the user palette graticule
DISplay:COLOr:PALEtte:USEr:HIStogram	Sets or returns the user palette histogram color
DISplay:COLOr:PALEtte:USEr:MASK	Sets or returns the user palette mask color
DISplay:COLOr:PALEtte:USEr:MASKHighlight	Sets or returns the user palette mask hits color
DISplay:COLOr:PALEtte:USEr:MATH<x>	Sets or returns the user palette math colors
DISplay:COLOr:PALEtte:USEr:REF<x>	Sets or returns the user palette reference colors
DISplay:COLOr:REFCOLOr	Sets or returns the color to be used for reference traces
DISplay:DESKew	Sets or queries the state of the Display Only button
DISplay:FILTer	Sets or returns the type of interpolation to use for the display
DISplay:FORMat	Sets or returns the display format
DISplay:GRAticule	Sets or returns the type of graticule that is displayed
DISplay:INTENSITy?	Returns the waveform and graticule saturation levels
DISplay:INTENSITy:WAVEform:IMAGEView	Sets or returns the waveform saturation level for imageview waveforms
DISplay:INTENSITy:WAVEform:RECORDView	Sets or returns the waveform saturation level for recordview waveforms
DISplay:PERSistence	Sets or returns display persistence setting

Command	Description
DISplay:PERSistence:RESET	Sets or returns the clearing of persistence data
DISplay:SCREENText?	Return all screen text settings
DISplay:SCREENText:LABel<x>?	Returns the screen text setting for a given label
DISplay:SCREENText:LABel<x>:NAMe	Sets or returns the text to be displayed for a given label
DISplay:SCREENText:LABel<x>:XPOS	Sets or returns the horizontal position of a given label
DISplay:SCREENText:LABel<x>:YPOS	Sets or returns the vertical position of a given label
DISplay:SCREENText:STATE	Sets or returns 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
DISplay:STYle	Sets or returns data display style
DISplay:TRIGBar	Sets or returns the display setting of the trigger level indicator bar(s)
DISplay:TRIGT	Sets or returns the display of the trigger point indicator
DISplay:VARpersist	Sets or returns the persistence decay time
DISplay:WAVEform	Sets or returns 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.*

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

Command	Description
EMail:TRIGger	Sets or queries whether e-mail is sent when a trigger occurs
EMail:WAVEform	Sets or queries whether waveform data is included in the e-mail

File System Command Group

Use the commands in the File System Command Group to help you use the built-in hard disk drive and floppy 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.

Command	Description
FILESystem?	Returns the file system state
FILESystem:COPy	Copies one or more files to a new file
FILESystem:CWD	Sets or returns the current working directory for FILESystem GPIB commands
FILESystem:DELEte	Deletes a named file or directory

Command	Description
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.

Command	Description
EXPort	Copies a waveform to a specified file or returns formatting and file information
EXPort:FILEName	Sets or returns the export file path
EXPort:FORMat	Sets or returns the export image format
EXPort:IMAGe	Sets or returns the inksaver mode for exporting waveforms
EXPort:PALEtte	Sets or returns the export color palette
EXPort:READOuts	Sets or returns the position of the readouts for export
EXPort:VIEW	Sets or returns the export screen capture area
HARDCopy	Sends a screen copy to the selected port or returns the selected port and file path
HARDCopy:FILEName	Sets or returns the hard copy file path
HARDCopy:IMAGe	Sets or returns the inksaver mode for hard copy printing
HARDCopy:LAYout	Sets or returns the page orientation for hard copy
HARDCopy:PALEtte	Sets or returns the hard copy color palette
HARDCopy:PORT	Sets or returns whether the hard copy data will be sent to a file or printed on the next hard copy command

Command	Description
HARDCopy:READOuts	Sets or returns the position of the readouts for hardcopy
HARDCopy:VIEW	Sets or returns 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

Command	Description
HIStogram?	Return all histogram parameters
HIStogram:BOX	Sets or returns the left, top, right, and bottom positions of the histogram box, in source waveform coordinates
HIStogram:BOXPcnt	Sets or returns same as HIStogram:BOX, but in percentage coordinates, with 0,0 upper left and 100,100 lower right
HIStogram:COUNt	Clears histogram count source data and restarts counting
HIStogram:DATa?	Returns a comma-separated list of histogram data numbers
HIStogram:DISplay	Sets or returns whether histogram data is displayed and the current display type setting
HIStogram:FUNCtion	Sets the type of histogram to create Or returns the current histogram type
HIStogram:MODe	Sets type of histogram to create or turns histograms off Or returns the current histogram type or that histogram display is disabled
HIStogram:SIZe	Sets or returns the width (or height) of the histogram on the screen in divisions

Command	Description
HIStogram:SOUrce	Sets or returns which source waveform will be compared against the histogram box when the histogram testing is enabled
HIStogram:STATE	Sets or returns 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.

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:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed
HORizontal:MAIn?	Returns the time per division of the main time base
HORizontal[:MAIn]:DELay:MODe	Sets or returns the main time base trigger delay mode
HORizontal[:MAIn]:DELay:POSition	Sets or returns the main time base position when Horizontal Delay Mode is turned off
HORizontal[:MAIn]:DELay:TIMe	Sets or returns the main time base trigger delay time
HORizontal:MAIn:INTERPRatio?	Returns the main horizontal time base interpolation ratio
HORizontal[:MAIn]:POSition	Sets or returns the waveform horizontal position on the display
HORizontal:MAIn:SAMPLERate	Sets the horizontal sample rate to the desired number of samples per second Or returns the current horizontal sample rate

Command	Description
HORizontal[:MAIn]:SCAle	Sets time per division for the main time base Or returns the main time base horizontal scale
HORizontal:MAIn:UNIts	Sets or returns the units for the horizontal time base
HORizontal:MAIn:UNIts:STRing	Sets or returns the units string for the horizontal main time base trigger delay
HORizontal:RECOrdlength	Sets the horizontal record length to the number of data points in each frame Or returns the current horizontal record length
HORizontal:RESOlution	Sets the horizontal record length to the number of data points in each frame and simultaneously adjusts the sample rate to maintain a constant time/division Or returns the current horizontal record length
HORizontal:ROLL	Sets or returns the horizontal roll mode status
HORizontal:TIMEStamp:CH<x>?	Returns the absolute trigger date and time for the specified frame and waveform
HORizontal:TIMEStamp:REF<x>?	Returns the absolute trigger date and time for FastFrame reference

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.

Command	Description
TRIGger:A:CAN:CONDition	Sets or returns the CAN condition
TRIGger:A:CAN:DATa:DIRection	Sets or queries the CAN trigger condition to be valid on a READ, WRITE, or either
TRIGger:A:CAN:FORMat	Sets or queries the CAN data format
TRIGger:A:CAN:DATa:LEVel	Sets or queries the CAN Trigger threshold for the CAN data source
TRIGger:A:CAN:DATa:SOUrce	Sets or queries the CAN data source
TRIGger:A:CAN:DATa:VALue	Sets or quires the binary data string used for CAN Trigger if the trigger condition is ID or IDANDDATA
TRIGger:A:CAN:FRAMEtype	Sets or queries the CAN trigger frame type
TRIGger:A:CAN:IDENTifier:MODe	Sets or queries the CAN trigger identifier mode

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: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
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

Command	Description
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: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: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
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

Command	Description
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

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 email 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.

Command	Description
MASK?	Returns the states of all settable mask parameters
MASK:AUTOAdjust	Optimizes signal position within the mask to minimize hits
MASK:AUTOAdjust:HDELTA	Sets or returns how far autofit searches horizontally
MASK:AUTOAdjust:VDELTA	Sets or returns how far autofit searches vertically
MASK:AUTOSet:AUTOAdjust	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
MASK:AUTOSet:HPOS	Controls whether the autoset algorithm will attempt to change the horizontal position of the signal while attempting to autoset
MASK:AUTOSet:HSCAle	Controls whether the autoset algorithm will attempt to change the horizontal scale while attempting to autoset
MASK:AUTOSet:MODe	Controls whether or not an autoset will be automatically done after a standard mask is selected
MASK:AUTOSet:OFFSETAdj	Sets autoset to not enforce the rule that, for pulse standards, requires 0V 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
MASK:AUTOSet:STANdard	Selects the standard mask in autoset
MASK:AUTOSet:TRIGger	Controls if a MASK:AUTOSet affects the trigger level
MASK:AUTOSet:USER:ONE	Sets or returns the level 1 used by autoset for user masks
MASK:AUTOSet:USER:TYPe	Sets or returns how autoset scales a mask
MASK:AUTOSet:USER:ZERo	Sets or returns the level 0 used by autoset for user masks
MASK:AUTOSet:VPOS	Controls whether the autoset algorithm will affect the vertical position (or offset) of the signal while attempting to autoset
MASK:AUTOSet:VSCAle	Controls whether the autoset algorithm will attempt to change the vertical scale while attempting to autoset
MASK:COPy:USER	Copies the currently displayed mask to the User mask

Command	Description
MASK:COUNt	Resets to zero the number of hits for all mask segments on the specified source
MASK:COUNt:FAILURES?	Returns the number of pass/fail tests that have failed
MASK:COUNt:HITS?	Returns the total number of mask hits on all mask segments
MASK:COUNt:SEG<m>:HITS?	Returns the number of hits in mask segment <m>
MASK:COUNt:STATE	Sets or returns the mask hits count state
MASK:COUNt:TESTS?	Returns the number of pass/fail tests that have occurred
MASK:COUNt:TOTal?	Returns the sum of all hits in all mask segments
MASK:COUNt:VIOLATIONS?	Returns the number of test violations that have occurred in the current pass/fail test
MASK:COUNt:WAVEFORMS?	Returns the number of waveforms that have been acquired and processed during pass/fail testing
MASK:DISplay	Sets or returns whether or not defined masks are displayed on the screen
MASK:FILTer	Sets or returns whether the selected optical reference receiver mask filter is enabled
MASK:HIGHLIGHTHits	Sets or returns whether or not hits in a mask are highlighted in different colors than other waveform data
MASK:INVert	Controls whether the mask is drawn inverted
MASK:LOCk	Sets or returns the mask lock to waveform mode
MASK:MARgin:PERCent	Sets or queries the mask margin percentage
MASK:MARgin:STATE	Sets or returns the state of the mask margins
MASK:MASKPRE:AMPlitude	Sets or returns the current mask's nominal pulse amplitude in volts
MASK:MASKPRE:HSCAle	Sets or returns the nominal timing resolution used to draw the current mask, in time/division
MASK:MASKPRE:HTRIGPOS	Sets or returns the nominal trigger position (pulse leading edge) used to draw the current mask as a fraction of the display width
MASK:MASKPRE:PATTERNBITS	Sets or returns the number of bits used for serial trigger for the current mask standard
MASK:MASKPRE:PRESAMPBITS	Sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing

Command	Description
MASK:MASKPRE:RECOrdlength	Sets or returns the nominal record length for pulse mask testing
MASK:MASKPRE:TRIGTOSAMP	Sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
MASK:MASKPRE:VOFFSet	Sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels
MASK:MASKPRE:VPOS	Sets or returns the nominal vertical position, in divisions, used to vertically position the input channels
MASK:MASKPRE:VSCAle	Sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels
MASK:MASKPRE:WIDth	Sets or returns the nominal bit width in seconds
MASK:POLarity	Sets or returns 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 returns 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 returns the stop on violation setting
MASK:TESt:AUX:COMPLetion	Sets or returns the test:aux on pass/fail test completion mode
MASK:TESt:AUX:FAILure	Sets or returns the test:aux status on pass/fail test failure mode
MASK:TESt:BEEP:COMPLetion	Sets or returns the beep on pass/fail test completion mode
MASK:TESt:BEEP:FAILure	Sets or returns the beep status on pass/fail test failure mode
MASK:TESt:DELay	Sets or returns the amount of time, in seconds, to wait after starting pass/fail testing to evaluate waveforms
MASK:TESt:HARDCopy	Sets or returns the hard copy in pass/fail failure mode

Command	Description
MASK:TESt:LOG:FAILure	Sets or returns the log status on pass/fail test failure mode
MASK:TESt:REPeat	Sets or returns the repeat pass/fail test on completion mode
MASK:TESt:SAMple	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
MASK:TESt:SAMple:THReshold	Sets or returns the minimum number of hits in mask regions needed to cause the pass/fail status to change from PASSING to FAILING
MASK:TESt:SAVEWFM	Sets or returns the save waveform to file on failure setting
MASK:TESt:SAVEWFM:FILEName	Sets or returns the name of the directory and file to use with the MASK:TESt:SAVEWFM command
MASK:TESt:SRQ:COMPLetion	Sets or returns an SRQ on pass/fail test completion mode
MASK:TESt:SRQ:FAILure	Sets or returns the SRQ status on pass/fail test failure mode
MASK:TESt:STATE	Sets or returns the state of mask pass/fail testing
MASK:TESt:STATUS?	Returns the pass/fail test status
MASK:TESt:STOP:FAILure	Sets or returns the stop status on pass/fail test failure mode
MASK:TESt:THReshold	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
MASK:TESt:WAVEform	Sets or returns the number of waveforms to test during pass/fail testing
MASK:USER:AMPlitude	Sets or returns the current mask's nominal pulse amplitude in volts
MASK:USER:BITRate	Sets or returns the user mask's bit rate
MASK:USER:HSCAle	Sets or returns the nominal timing resolution used to draw the mask, in time/division
MASK:USER:HTRIGPOS	Sets or returns the nominal trigger position (pulse leading edge) used to draw the mask as a fraction of the display width
MASK:USER:LABel	Sets or returns the label assigned to a user mask
MASK:USER:PATTERNBITS	Sets or returns the number of bits used for serial trigger for the User mask standard

Command	Description
MASK:USER:PRESAMPBITS	Sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing
MASK:USER:RECOrdlength	Sets or returns 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
MASK:USER:SEG<m>:POINTS	Sets or returns the X-Y user coordinates of all points in the specified User mask segment
MASK:USER:TRIGTOSAMP	Sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
MASK:USER:VOFFSet	Sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels
MASK:USER:VPOS	Sets or returns the nominal vertical position, in divisions, used to vertically position the input channels
MASK:USER:VSCAle	Sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels
MASK:USER:WIDth	Sets or returns the nominal bit width in seconds
RECAll:MASK	Recalls a stored mask as the User mask
SAVe:MASK	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.

Command	Description
MATH<x>?	Returns the specified math waveform settings
MATH<x>:DEFine	Sets or returns the math<x> waveform definition for the specified waveform
MATH<x>:FILTer:MODe	Sets or returns the filter risetime parameter (centered or shifted)
MATH<x>:FILTer:RISetime	Sets or returns the filter risetime parameter
MATH<x>:LABel:NAMe	Sets or returns the label string used for annotating the displayed math waveform
MATH<x>:LABel:XPOS	Sets or returns the X screen offset where the math waveform label is displayed
MATH<x>:LABel:YPOS	Sets or returns the Y screen offset where the math waveform label is displayed
MATH<x>:NUMAVg	Sets or returns the acquisition number at which the averaging algorithm will begin exponential averaging
MATH<x>:SPECTral?	Returns the specified math waveform spectral setups
MATH<x>:SPECTral:CENTER	Sets or returns the center frequency of the spectral analyzer output data span
MATH<x>:SPECTral:GATEPOS	Sets or returns the position of the center of the gate
MATH<x>:SPECTral:GATEWIDTH	Sets or returns the gate width input to the spectral analyzer
MATH<x>:SPECTral:LOCk	Sets or returns the state of the spectral locking for the specified math waveform
MATH<x>:SPECTral:MAG	Sets or returns the units of the SpectralMag function in the specified math string
MATH<x>:SPECTral:PHASE	Sets or returns the units of the SpectralPhase function in the specified math string
MATH<x>:SPECTral:REFLevel	Sets or returns the vertical position of the output data from the spectral analyzer

Command	Description
MATH<x>:SPECTral:REFLEVELOffset	Sets or returns the spectral reference level offset used for calculating the dB value
MATH<x>:SPECTral:RESBw	Sets or returns the resolution bandwidth of the spectral analyzer
MATH<x>:SPECTral:SPAN	Sets the top of the frequency span to a value that is closest to the specified value Or returns the specified math waveform's current frequency span value
MATH<x>:SPECTral:SUPPress	Sets or returns the phase suppression threshold
MATH<x>:SPECTral:UNWRap	Enables or disables or returns the state of the phase unwrap of the spectral analyzer output data
MATH<x>:SPECTral:WINdow	Sets or returns the window function used to multiply the input data to the spectral analyzer
MATH<x>:UNITString	Sets or returns the units used for the math waveform; command overrides the default unit string
MATH<x>:VERTical:POSition	Sets or returns the specified math waveform vertical position
MATH<x>:VERTical:SCAle	Sets or returns the specified math waveform vertical scale (per div)
MATHArbflt<x>:FILepath	Sets the file path for a file of filter coefficients and reads the file
MATHArbflt<x>:READFile	Forces a reread of the filter file for the specified arbitrary filter
MATHVAR?	Sets or returns all 8 numerical values you can use within math expressions
MATHVAR:VAR<x>	Sets or returns a numerical value you can use within math expressions

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.

Command	Description
MEASUrement?	Returns all measurement parameters
MEASUrement:GATing	Sets or returns 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 returns the search direction to use for immediate delay measurements
MEASUrement:IMMed:DELay:EDGE[1]	Sets or returns the slope of the edge used for immediate delay "from" waveform measurements
MEASUrement:IMMed:DELay:EDGE2	Sets or returns 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 returns the location of the noise measurement on the eye waveform
MEASUrement:IMMed:REFLevel?	Returns the reference levels
MEASUrement:IMMed:REFLevel:ABSolute:HIGH	Sets or returns the high reference level
MEASUrement:IMMed:REFLevel:ABSolute:LOW	Sets or returns the low reference level
MEASUrement:IMMed:REFLevel:ABSolute:MID<x>	Sets or returns the middle reference levels
MEASUrement:IMMed:REFLevel:METHod	Sets or returns the method used to calculate measurements
MEASUrement:IMMed:REFLevel:PERCent:HIGH	Sets or returns the top level for rise time (90% level)
MEASUrement:IMMed:REFLevel:PERCent:LOW	Sets or returns the low level for rise time (10% level)
MEASUrement:IMMed:REFLevel:PERCent:MID<x>	Sets or returns the mid levels for measurements

Command	Description
MEASUrement:IMMed:SOUrce<x>	Sets or returns the source to measure "to" or “from” for phase or delay immediate measurements
MEASUrement:IMMed:SOUrce<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source
MEASUrement:IMMed:TYPe	Sets or returns 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 returns the search direction to use for delay measurements
MEASUrement:MEAS<x>:DELay:EDGE<x>	Sets or returns 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 returns the method used to calculate the 0% and 100% reference level
MEASUrement:MEAS<x>:MINImum?	Returns the minimum value found since the last statistical reset
MEASUrement:MEAS<x>:NOISe	Sets or returns 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 returns the top level for rise time (90% level)
MEASUrement:MEAS<x>:REFLevel:ABSolute:LOW	Sets or returns the low level for rise time (10%)
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID<x>	Sets or returns the mid1 or mid2 level for measurements
MEASUrement:MEAS<x>:REFLevel:METHod	Sets or returns the reference level units used for measurement calculations
MEASUrement:MEAS<x>:REFLevel:PERCent:HIGH	Sets or returns the top level for rise time (90% level)

Command	Description
MEASUrement:MEAS<x>:REFLevel:PERCent:LOW	Sets or returns the low level for rise time (10% level)
MEASUrement:MEAS<x>:REFLevel:PERCent:MID<x>	Sets or returns the mid1 or mid2 level for delay measurements
MEASUrement:MEAS<x>:SOUrce<x>	Sets or returns the source to measure "from" or to" for phase or delay measurements
MEASUrement:MEAS<x>:SOUrce<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source for the specified measurement slot
MEASUrement:MEAS<x>:STATE	Sets or returns 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 returns 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 returns the method used for calculating reference levels
MEASUrement:NOISe	Sets or returns the location of the noise measurement on the eye waveform
MEASUrement:REFLevel:ABSolute:HIGH	Sets or returns the top reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW	Sets or returns the low reference level for rise time
MEASUrement:REFLevel:ABSolute:MID<x>	Sets or returns the mid1 or mid2 reference level for measurements
MEASUrement:REFLevel:METHod	Sets or returns the method for assigning high and low reference levels
MEASUrement:REFLevel:PERCent:HIGH	Sets or returns the top reference percent level for rise time
MEASUrement:REFLevel:PERCent:LOW	Sets or returns the low reference percent level for rise time
MEASUrement:REFLevel:PERCent:MID<x>	Sets or returns the mid1 or mid 2 reference percent level for waveform measurements
MEASUrement:SOUrce<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source
MEASUrement:STATIstics:COUNt	Clears existing measurement statistics from memory

Command	Description
MEASUrement:STATIstics:MODe	Turns measurement statistics on or off and sets or returns which pair of statistical accumulations is displayed
MEASUrement:STATIstics:WEIghting	Sets or returns 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.

Command	Description
*DDT	Sets or returns the commands that will be executed by the group execute trigger
*IDN?	Returns the instrument identification code
*LRN?	Returns a listing of instrument settings
*TRG	Performs the group execute trigger (GET)
*TST?	Tests the GPIB interface and returns status
AUTOSet	Sets the vertical, horizontal and trigger controls to provide a stable display of the selected waveform
AUXout?	Returns the auxiliary out setup
AUXout:EDGE	Sets or returns the direction in which the trigger output signal will transition when a trigger occurs
AUXout:SOUrce	Sets or returns the trigger source at the BNC connection
BELl	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 returns the date that the instrument can display
HDR	Sets or returns the Response Header Enable State
HEADer	Sets or returns the Response Header Enable State

Command	Description
ID?	Returns identifying information about the instrument and its firmware
LOCk	Sets or returns the front panel lock state
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:SOUrce	Sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source
ROSc:STATE?	Returns whether the time base reference oscillator is locked
	Sets or returns the selected frequency for the timebase reference output signal
SET?	Returns a listing of instrument settings
TEKSecure	Initializes both waveform and setup memories
TIME	Sets or returns the time displayed by the instrument
UNLock	Unlocks front panel
VERBose	Sets or returns 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.

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

Command	Description
FACtory	Resets the instrument to factory default settings
RECAll:SETUp	Recalls saved instrument settings
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 returns the format for saved waveforms
SETUp:NAMe	Sets or returns the user-defined setup label

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.

Command	Description
*CLS	Clears status
*ESE	Sets or returns 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 returns the power on status flag
*PUD	Sets or returns a string of protected user data
*RST	Resets the instrument to factory default settings
*SRE	Sets or returns the bits in the Service Request Enable Register
*STB?	Returns the contents of the Status Byte Register

Command	Description
*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 returns 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?	Return number of events in the event queue

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.

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 returns the type of coupling for the edge trigger

Command	Description
TRIGger:{A|B}:EDGE:SLOpe	Sets or returns the slope for the edge trigger
TRIGger:{A|B}:EDGE:SLOpe:AUX	Sets or returns the slope for the edge trigger for the AUX In input
TRIGger:{A|B}:EDGE:SLOpe:CH<x>	Sets or returns the slope for the edge trigger for the channel
TRIGger:{A|B}:EDGE:SOUrce	Sets or returns the source for the edge trigger
TRIGger:{A|B}:LEVel	Sets or returns the level for the trigger
TRIGger:{A|B}:LEVel:CH<x>	Sets or returns 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:CLAss	Sets or returns the type of trigger logic
TRIGger:{A|B}:LOGIc:FUNCtion	Sets or returns 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:CH<x>	Sets or queries the logical input for the logic trigger channel specified by x
TRIGger:{A|B}:LOGIc:PATtern:INPut:CH<x>	Sets or returns the logic pattern trigger input for the indicated channel
TRIGger:{A|B}:LOGIc:PATtern:WHEn	Sets or returns the condition for generating the logic pattern trigger
TRIGger:{A|B}:LOGIc:PATtern:WHEn:LESSLimit	Sets or returns the maximum time that the selected pattern may be true and still generate a logic pattern trigger
TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit	Sets or returns 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 returns the clock edge polarity for setup and hold triggering
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:SOUrce	Sets or returns the clock source for the logic trigger setup and hold input
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold	Sets or returns the clock voltage threshold for setup and hold trigger
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold:CH<x>	Sets or returns 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

Command	Description
	Sets or returns the clock level for the setup and hold trigger
	Sets or returns the data level for the setup and hold trigger
TRIGger:{A|B}:LOGIc:SETHold:DATa:SOUrce	Sets or returns the data source for the setup and hold trigger
TRIGger:{A|B}:LOGIc:SETHold:DATa:THReshold	Sets or returns the data voltage threshold for setup and hold trigger
TRIGger:{A|B}:LOGIc:SETHold:DATa:THReshold:CH<x>	Sets or returns 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 returns the hold time for the setup and hold violation triggering
TRIGger:{A|B}:LOGIc:SETHold:QUAlify	Sets or returns the setup and hold violation trigger qualifications
TRIGger:{A|B}:LOGIc:SETHold:SETTime	Sets or returns 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:INPut:CH<x>	Sets or returns the slope for indicated channel of the logic state trigger
TRIGger:{A|B}:LOGIc:STATE:WHEn	Sets or returns 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
TRIGger:{A|B}:LOGIc:THReshold:CH<x>	Sets or returns the logic state trigger threshold voltage for the specified channel
TRIGger:{A|B}:LOWerthreshold:CH<x>	Sets or returns 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:CLAss	Sets or returns the type of pulse on which to trigger
	Sets or queries the acceptance/rejection of the glitch pulse trigger
TRIGger:{A|B}:PULse:GLItch:POLarity	Sets or returns the polarity for the pulse glitch trigger
TRIGger:{A|B}:PULse:GLItch:POLarity:CH<x>	Sets or returns the polarity for the A or B pulse glitch trigger for the channel
TRIGger:{A|B}:PULse:GLItch:QUAlify	Sets or returns the glitch trigger qualifications

Command	Description
TRIGger:{A|B}:PULse:GLItch:TRIGIF	Sets or returns the acceptance or rejection of pulse glitch trigger, based on width
TRIGger:{A|B}:PULse:GLItch:WIDth	Sets or returns 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 returns the polarity for the pulse runt trigger
TRIGger:{A|B}:PULse:RUNT:POLarity:CH<x>	Sets or returns the polarity for the A or B pulse runt trigger for the channel
TRIGger:{A|B}:PULse:RUNT:QUAlify	Sets or returns 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 returns the upper limit for the A pulse runt trigger
TRIGger:{A|B}:PULse:RUNT:THReshold:LOW	Sets or returns 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 returns the type of pulse width the trigger checks for when it uncovers a runt
TRIGger:{A|B}:PULse:RUNT:WIDth	Sets or returns 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 returns the source for the A pulse trigger
TRIGger:{A|B}:PULse:TIMEOut:POLarity	Sets or returns the polarity for the A pulse timeout trigger
TRIGger:{A|B}:PULse:TIMEOut:POLarity:CH<x>	Sets or returns the polarity for the A or B pulse timeout trigger for the channel
TRIGger:{A|B}:PULse:TIMEOut:QUAlify	Sets or returns the Timeout trigger qualifications
TRIGger:{A|B}:PULse:TIMEOut:TIMe	Sets or returns 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 returns the delta time used in calculating the transition value
TRIGger:{A|B}:PULse:TRANsition:POLarity	Sets or returns the polarity for the A pulse transition trigger
TRIGger:{A|B}:PULse:TRANsition:POLarity:CH<x>	Sets or returns the polarity for the A or B pulse transition trigger for the channel

Command	Description
TRIGger:{A|B}:PULse:TRANsition:QUAlify	Sets or returns 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 returns the upper transition trigger threshold
TRIGger:{A|B}:PULse:TRANsition:THReshold:LOW	Sets or returns 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
TRIGger:{A|B}:PULse:WIDth:HIGHLimit	Sets or returns the upper limit for the A pulse width trigger
TRIGger:{A|B}:PULse:WIDth:LOWLimit	Sets or returns the lower limit for the pulse width trigger
TRIGger:{A|B}:PULse:WIDth:POLarity	Sets or returns the polarity for the pulse width trigger
TRIGger:{A|B}:PULse:WIDth:POLarity:CH<x>	Sets or returns the polarity for the A or B pulse width trigger for the channel
TRIGger:{A|B}:PULse:WIDth:QUAlify	Sets or returns the pulse width trigger qualifications
TRIGger:{A|B}:PULse:WIDth:WHEn	Sets or returns 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:EVENT	Sets or queries the window trigger event
TRIGger:{A|B}:PULse:WINdow:QUAlify	Sets or returns the window trigger qualifications
TRIGger:{A|B}:PULse:WINdow:THReshold:BOTh	Sets both window thresholds
TRIGger:{A|B}:PULse:WINdow:THReshold:HIGH	Sets or returns high window threshold
TRIGger:{A|B}:PULse:WINdow:THReshold:LOW	Sets or returns 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 returns minimum width of violation

Command	Description
TRIGger:{A|B}:PULse:WINdow?	Returns the current window pulse trigger parameters
TRIGger:{A|B}:PULse?	Returns the pulse trigger parameters
TRIGger:{A|B}:UPPerthreshold:CH<x>	Sets or queries the CH<x> upper trigger level for TRIGger:LVLSrcpreference
TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:PULSEForm	Sets or returns 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 returns the communication bit rate
TRIGger:A:COMMunication:CLOCk:POLarity	Sets or returns 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 returns the type of holdoff for the A trigger
TRIGger:A:HOLDoff:TIMe	Sets or returns the A trigger holdoff time
TRIGger:A:HOLDoff?	Returns the A trigger holdoff parameters
TRIGger:A:LOGIc?	Returns all of the A trigger logic parameters
TRIGger:A:MODe	Sets or returns the A trigger mode
TRIGger:A:TYPe	Sets or returns the type of A trigger
TRIGger:A:VIDeo:CUSTom?	Returns the custom video parameters for the A trigger
TRIGger:A:VIDeo:CUSTom:FORMat	Sets or returns the video scan format when the video standard is set to Custom
TRIGger:A:VIDeo:CUSTom:LINEPeriod	Sets or returns the custom video line period
TRIGger:A:VIDeo:CUSTom:SYNCInterval	Sets or returns the custom video sync interval
TRIGger:A:VIDeo:FIELD	Sets or returns the video field trigger
TRIGger:A:VIDeo:HOLdoff:FIELD	Sets or returns the trigger holdoff in video field units

Command	Description
TRIGger:A:VIDeo:LINE	Sets or returns the trigger delay as a number of video lines
TRIGger:A:VIDeo:POLarity	Sets or returns the polarity of the video trigger
TRIGger:A:VIDeo:SCAN	Sets or returns the video scan rate
TRIGger:A:VIDeo:SOUrce	Sets or returns the video trigger source
TRIGger:A:VIDeo:STANdard	Sets or returns the video standard
TRIGger:A:VIDeo?	Returns the video parameters for the A trigger
TRIGger:AUXLevel	Sets or returns the level for the auxiliary trigger
TRIGger:B	Sets the B trigger level to 50% or returns the B trigger parameters
TRIGger:B:BY	Sets or returns B trigger time or event qualifiers
TRIGger:B:EVENTS:COUNt	Sets or returns 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 returns the source for the trigger reset
TRIGger:B:RESET:STATE	Sets or returns the state of the B trigger
TRIGger:B:RESET:THReshold	Sets or returns the threshold for the trigger reset
TRIGger:B:RESET:TIMEOut	Sets or returns the reset time for the timeout trigger reset
TRIGger:B:RESET:TRANsition	Sets or returns the type of transition required for a transition trigger reset
TRIGger:B:RESET:TYPe	Sets or returns the type of A→B sequential trigger reset
TRIGger:B:TIMe	Sets or returns the B trigger delay time
TRIGger:B:STATE	Returns the current state of the triggering system
TRIGger:B:TYPe	Sets or returns the type of B trigger
TRIGger:LVLSrcpreference	Sets or returns how the trigger source and levels interact
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.

Command	Description
AUXIn:BANdwidth	Sets or returns the selectable low-pass bandwidth limit filter of the AUX In input
AUXIn:COUPling	Sets or returns the input attenuator coupling setting for the AUX In input
AUXIn:OFFSet	Sets or returns 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 returns 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
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:UNIts?	Returns a string describing the units of measure for the probe attached to the AUX In input
AUXIn:PROBEFunc:EXTAtten	Sets or returns the input-output ratio, of external attenuation or gain, between the signal and the AUX In input
AUXIn:PROBEFunc:EXTDBatten	Sets or returns 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 returns the unit of measurement for the external attenuator of the AUX In input

Command	Description
CH<x>?	Returns vertical parameters for the specified channel
CH<x>:BANdwidth	Sets or returns the bandwidth of the specified channel
CH<x>:COUPling	Sets or returns the coupling setting for the specified channel
CH<x>:DESKew	Sets or returns the deskew time for the specified channel
CH<x>:LABel:NAMe	Defines or returns the label for the channel waveform
CH<x>:LABel:XPOS	Sets or returns the X display coordinate for the channel waveform label
CH<x>:LABel:YPOS	Sets or returns the Y display coordinate for the channel waveform label
CH<x>:OFFSet	Sets or returns the channel offset
CH<x>:POSition	Sets or returns 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 returns the default range of the probe
CH<x>:PRObe:GAIN?	Returns the gain of the probe that is attached to the specified channel
CH<x>:PRObe:ID?	Returns the type and 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:ID:SERnumber?	Returns the serial number of the probe that is attached to the specified channel
CH<x>:PRObe:RANge	Sets or returns 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:UNIts?	Returns the units of the probe that is attached to the specified channel
CH<x>:PROBECal?	Returns the probe calibration status

Command	Description
CH<x>:PROBEFunc:EXTAtten	Sets the attenuation value for the specified channel to the specified scale factor Or returns the user-specified attenuation
CH<x>:PROBEFunc:EXTDBatten	Sets the attenuation value for the specified channel to the specified value, in decibels Or returns the user-specified attenuation, in decibels
CH<x>:PROBEFunc:EXTUnits	Sets the unit of measurement for the external attenuator of the specified channel Or returns the user-specified unit of measurement for the external attenuator
CH<x>:SCAle	Sets or returns the vertical scale of the specified channel
CH<x>:TERmination	Sets channel input termination
REF<x>:HORizontal:POSition	Sets or returns 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 returns the label of the designated waveform
REF<x>:LABel:XPOS	Sets or returns the position of the reference waveform label on the X axis
REF<x>:LABel:YPOS	Sets or returns the position of the reference waveform label on the Y axis
REF<x>:VERTical:POSition	Sets or returns the vertical position of the specified reference waveform
REF<x>:VERTical:SCAle	Sets or returns 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 returns whether the specified channel is on or off
SELect:CONTROl	Sets or returns the waveform controlled by the front panel
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

Command	Description
CH<x>:BANdwidth:ENHanced	Sets or queries the state of bandwidth enhancement
CH<x>:BANdwidth:ENHanced:APPLYtoall	Sets or queries whether the enhanced bandwidth filter applies to all active channels
CH<x>:BANdwidth:ENHanced:STATE?	This query returns the state of the bandwidth enhancement filter
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
SELect:CH<x>	Sets or queries the displayed state of the specified channel waveform

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.

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 WFMInpre and WFMOutpre commands for more information.

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

Transferring Waveform Data from the Instrument.

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

1. Select the waveform source(s) using DATa:SOUrce.
2. Specify the waveform data format using DATa:ENCdg.
3. Specify the number of bytes per data point using WFMOutpre:BYT_Nr.

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

4. Specify the portion of the waveform that you want to transfer using DATa:STARt and DATa:STOP.
5. Transfer waveform preamble information using WFMOutpre.
6. Transfer waveform data from the instrument using CURVe.

Transferring Waveform Data to the Instrument.

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

Note: FastAcq waveforms cannot be returned to the instrument via the CURVe command.

Command	Description
ALLOcate:WAVEform:REF<x>?	Returns the record length for the specified reference waveform, if active, or zero (0) if not active
CURVe	The command format transfers waveform data to the instrument (reference memory location specified by DATa:DESTination) The query format transfers waveform data from instrument specified by the DATa:SOUrce command

Command	Description
CURVENext?	Returns unique waveform data from a new acquisition
CURVEStream	Continuously transfers waveform data from the instrument as it is acquired
DATa	Sets the format and location of the waveform data that is transferred with the CURVe Command Or returns the format and location of the waveform data that is transferred with the CURVe? command
DATa:DESTination	Sets or returns the reference memory location for storing waveform data sent to the instrument
DATa:ENCdg	Sets or returns the format of outgoing waveform data
DATa:SOUrce	Sets or returns the location of waveform data transferred from the instrument
DATa:STARt	Sets or returns the starting point in waveform transfer
DATa:STOP	Sets or returns the ending data point in waveform transfer
WAVFrm?	Returns WFMOutpre? and CURVe? data for the waveform as specified by the DATA:SOUrce command
WAVFRMStream?	Returns WFMOUTPRE? and CURVESTREAM? data for the waveforms specified by the DATASOURCE command
WFMInpre?	Returns the waveform formatting specification to be applied to the next incoming CURVE command data
WFMInpre:BIT_Nr	Sets or returns the number of bits per binary waveform point
WFMInpre:BN_Fmt	Sets or returns the format of binary data for the incoming waveform
WFMInpre:BYT_Nr	Sets or returns the binary field data width for the first ordered waveform
WFMInpre:BYT_Or	Sets or returns the byte order of waveform points
WFMInpre:ENCdg	Sets or returns the type of encoding for incoming waveform data
WFMInpre:NR_Pt	Sets or returns the number of points in the transmitted waveform record
WFMInpre:PT_Fmt	Sets or returns the point format of incoming waveform data

Command	Description
WFMInpre:PT_Off	Sets or returns the trigger point within the incoming waveform data
WFMInpre:WFId	Provided to allow a waveform extracted from the instrument to be more easily sent back
WFMInpre:XINcr	Sets or returns the horizontal sampling interval between incoming waveform points
WFMInpre:XUNit	Sets or returns the horizontal units of the incoming waveform
WFMInpre:XZEro	Sets or returns the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMInpre:YMUlt	Sets or returns the vertical scale factor, per digitizing level, of the incoming waveform points
WFMInpre:YOFf	Sets or returns the vertical position of the incoming waveform in digitizing levels
WFMInpre:YUNit	Sets or returns the vertical units of the incoming waveform
WFMInpre:YZEro	Sets or returns the offset of the incoming waveform
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command
WFMOutpre:BIT_Nr	Sets or returns the number of bits per waveform point that outgoing waveforms contain
WFMOutpre:BN_Fmt	Sets or returns the format of binary data for the waveform
WFMOutpre:BYT_Nr	Sets or returns the data width for the waveform
WFMOutpre:BYT_Or	Sets or returns the byte order of waveform points
WFMOutpre:ENCdg	Sets or returns the type of encoding for outgoing waveforms
WFMOutpre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query
WFMOutpre:PT_Fmt?	Returns the point format for the waveform
WFMOutpre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform
WFMOutpre:PT_ORder?	Returns whether the source waveform is DPO

Command	Description
WFMOutpre:WFId?	Returns a string describing the acquisition parameters for the waveform
WFMOutpre:XINcr?	Returns the horizontal sampling interval
WFMOutpre:XUNit?	Returns the horizontal units for the waveform
WFMOutpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger
WFMOutpre:YMUlt?	Returns the vertical scale factor per digitizing level for the waveform
WFMOutpre:YOFf?	Returns the vertical offset in digitizing levels for the waveform
WFMOutpre:YUNit?	Returns the vertical units for the waveform
WFMOutpre:YZEro?	Returns the vertical offset for the waveform

Zoom Command Group

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

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.

Command	Description
ZOOm	Sets the zoom transforms to default values for either live traces or all traces Or returns the current vertical and horizontal positioning and scaling of the display
ZOOm:GRAticule:SIZE	Sets or returns the size of the zoom graticule
ZOOm:MODe	Sets or returns the zoom mode
ZOOm:SCROLL:DIREction	Sets or returns the direction of AutoScroll for zoomed waveforms
ZOOm:SCROLL:LOCk	Sets or returns the state of Scroll Lock, on or off
ZOOm:SCROLL:SPEED	Sets or returns the speed of AutoScroll for zoomed waveforms
ZOOm:ZOOM<x>	Sets or returns zoom factors for the specified zoom
ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition	Sets or returns the horizontal zoom position for the specified waveform in the specified zoom
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAle	Sets or returns the horizontal zoom scale of the specified waveform in the specified zoom

Command	Description
ZOOm:ZOOM<x>:<wfm>:HORizontal:TIMEPosition	Sets or queries the zoom horizontal position of the specified waveform for the specified zoom in time
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition	Sets or returns the vertical zoom position for the specified waveform in the specified zoom
ZOOm:ZOOM<x>:<wfm>:VERTical:SCAle	Sets or returns the vertical zoom scale for the specified waveform in the specified zoom
ZOOm:ZOOM<x>:SCROLLLock	Sets or returns Scroll Lock for the specified zoom
ZOOm:ZOOM<x>:STATE	Specifies or returns a trace as zoomed, on or off

Commands Listed in Alphabetical Order

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:NUMAVg command.

`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.

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

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

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

ACQuire: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.*

Group Acquisition

Syntax

```
ACQuire:NUMEnv {<NR1>|INFInite}
ACQuire:NUMEnv?
```

Arguments

NR1 is the number of waveform acquisitions to be enveloped.

Examples

ACQUIRE:NUMENV 10 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: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 Waverform 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:NUMSAMPLES 5000 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:SAMPLINGMODE IT 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 ACQUIRE:STATE.

Group

Acquisition

Syntax

```
ACQuire:STATE {OFF|ON|RUN|STOP|<NR1>}
ACQuire:STATE?
```

Related Commands

ACQuire:STOPAfter

Arguments

OFF stops acquisitions.

STOP stops acquisitions.

ON starts acquisitions.

RUN starts 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.

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:NUMACq?, 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.

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

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

`ALIAS ?` might return `:ALIAS:STATE 1` indicating that the alias feature is on.

ALIas:CATalog? (Query Only)

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

Group

Alias

Syntax

```
ALIas:CATalog?
```

Examples

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

ALIas:DEFine

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALIas:STATE 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 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 autoset 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 autoset 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` autosets 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 autoset.

`VIDeo` autosets the displayed waveform.

`VLines` autosets the displayed waveform.

`VFields` autosets 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. 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:BANdwidth {TWEnty|ONEfifty|TWOfifty|FULl}
AUXIn:BANdwidth?
```

Arguments

`TWEnty` sets the upper bandwidth limit to 20 MHz.

`ONEfifty` sets the upper bandwidth limit to 150 MHz.

`TWOFIFTY` argument sets the upper bandwidth limit to 250 MHz.

`FULl` disables any optional bandwidth limiting. The AUX In input operates at its maximum bandwidth.

Examples

`AUXIN:BANDWIDTH TWENTY` sets the bandwidth of AUX In input to 20 MHz.

`AUXIN:BANDWIDTH?`, for DPO7054 instruments, might return `:AUXIN:BANDWIDTH 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. This command is equivalent to selecting Coupling from the Vertical menu.

Group

Vertical

Syntax

```
AUXIn:COUPling {AC|DC|GND|DCREJect}
AUXIn:COUPling?
```

Arguments

`AC` sets the AUX In input to AC coupling.

`DC` sets the AUX In input to DC coupling.

`GND` sets the AUX In input to ground. Only a flat, ground-level waveform is displayed.

`DCREJect` sets DC Reject coupling when a probe with that feature is attached to the auxiliary trigger input.

Examples

`AUXIN:COUPLING GND` sets the AUX In input to ground.

`AUXIN:COUPling?` might return `:AUXIN:COUPling DC` indicating that AUX In input is set to DC 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`.

AUXIn:PRObe:DEGAUSS (No Query Form)

This command starts a degauss cycle of the probe attached to the AUX In input.

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?
```

Arguments

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 default range of the probe, or the command queries the default range of the probe.

Group Vertical

Syntax

```
AUXIn:PRObe:FORCEDRange
AUXIn:PRObe:FORCEDRange?
```

Examples

AUXIN:PROBE:FORCEDRANGE sets the probe attached to the AUX In input to the default range of the probe.

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”.

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”.

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: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 AUX In input is 10 MΩ.

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 below:

`ATTENUATION 5X` sets the P7260 probe to ±0.75 V dynamic range with 6 GHz bandwidth and 5X attenuation.

`ATTENUATION 25X` sets the P7260 probe to ±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 ±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 ±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 ±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:SIGNAL PASS` opens a relay passing your signal to the instrument.

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

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.

Group Vertical

Syntax

```
AUXIn:PRObe:UNIts?
```

Examples `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 is used to specify 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 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

Use this command to specify 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 to 200.00 dB.

Examples

`AUXIN:PROBEFUNC:EXTDBATTEN 2.5` T specifies an external 2.5 dB attenuator on the AUX In input.

`AUXIN:PROBEFUNC:EXTDBATTEN?` might return `:AUXIN:PROBEFUNC:EXTDBATTEEN 2.5000E+00` indicating that the attenuation is 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
AUXIn:PROBEFunc:EXTUnits?
```

Related Commands

AUXIn:PRObe:UNIts?

Arguments

`<str>` 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` ? might return `:AUXOUT:EDGE FALLING` indicating a falling edge is produced when a trigger occurs.

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.

Group Miscellaneous

Syntax

```
AUXout:SOUrce {ATRIGger|BTRIGger|MAIn|DELayed|EVENT|REFOut}
AUXout:SOUrce?
```

Related Commands AUXout?

Arguments

`ATRIGger` sets the source at the BNC connector to the main trigger.

`BTRIGger` sets the source at the BNC connector to the delayed trigger.

`MAIn` sets the source at the BNC connector to the main trigger.

`DELayed` sets the source at the BNC connector to the delayed trigger.

`EVENT` sets the source at the BNC connector to a specified event.

`REFOut` sets the source at the BNC connector to the reference output.

Examples

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

BELl (No Query Form)

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

Group

Miscellaneous

Syntax

```
BELl
```

Examples

`BELL` is accepted but does nothing.

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 one of the commands listed in the table below.

Commands that affect BUSY? response

Operation	Command
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN or ACQuire:STATE1 (when ACQuire:STOPAfter is set to SEQuence)
Hard copy operation	HARDCopy STArt
Calibration step	Refer to the instrument Service Manual.

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.

Group Calibration

Syntax

```
CALibrate?
```

Examples

`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.*

Group

Calibration

Syntax

```
CALibrate:CALProbe:CH<x>?
```

Related Commands

CALibrate:PRObestate:CH<x>?, CH<x>:PROBECal?

Returns

H -1
Indicating the probe calibration failed for the selected channel.
H 0
Indicating the probe calibration passed for the selected channel.
H 1
Indicating the probe calibration is initialized for the selected channel.

Examples

`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:INTERNal`

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:INTERNal 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`.

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.

Group

Vertical

Syntax

```
CH<x>?
```

Examples

`CH1?` might return the following vertical parameters for channel 1: `:CH1:BANDWIDTH 1.0000E+09;COUPLING DC;DESKEW0.0000E+00;OFFSET 0.0000E+00;POSITION 0.0000E+00;SCALE 5.0000E-01;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`.

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 maximum bandwidth of the channel.

Available arguments depend upon the instrument and the attached accessories.

Group

Vertical

Syntax

```
CH<x>:BANdwidth {TWEnty|ONEfifty|TWOfifty|FIVe|FULl|<NR3>}
CH<x>:BANdwidth?
```

Arguments

`TWEnty` sets the upper bandwidth limit to 20 MHz.

`TWOfifty` sets the upper bandwidth limit to 250 MHz.

`ONEfifty` sets the upper bandwidth limit to 150 MHz.

`FIVe` argument sets the upper bandwidth limit to 500 MHz.

`FULl` disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.

<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

DSA70000 and DPO70000 Series only: This command sets or queries the state of bandwidth enhancement.

Group

Vertical

Syntax

```
CH<x>:BANdwidth:ENHanced {AUTO|OFF}
CH<x>:BANdwidth:ENHanced?
```

Arguments

AUTO allows bandwidth filtering if possible.

OFF prevents filtering.

Examples

CH1:BANDWIDTH:ENHANCED AUTO allows bandwidth enhancement filtering if possible.

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

CH<x>:BANdwidth:ENHanced:APPLYtoall

DSA70000 and DPO70000 Series only: This command sets or queries whether the enhanced bandwidth filter applies to all active channels.

Group

Vertical

Syntax

```
CH<x>:BANdwidth:ENHanced:APPLYtoall {ON|OFF}
CH<x>:BANdwidth:ENHanced:APPLYtoall?
```

Arguments ON sets all channels to use bandwidth enhancement filters.

OFF allows each channel to use different or no bandwidth enhancement filters.

Examples `CH1:BANDWIDTH:ENHANCED:APPLYTOALL ON` sets all active channels to use the same bandwidth enhancement filter.

`CH:BANDWIDTH:ENHANCED:APPLYTOALL OFF` allows each channel to use a different or no bandwidth enhancement filter.

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

DSA70000 and DPO70000 Series only: This query-only command returns the state of the bandwidth enhancement filter.

Group Vertical

Syntax

```
CH<x>:BANdwidth:ENHanced:STATE?
```

Examples `CH1:BANDWIDTH:ENHANCED:STATE?` might return `:CH1:BANDWIDTH:ENHANCED:STATE 1` indicating the bandwidth enhancement filter is on.

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.*

Group Vertical

Syntax

```
CH<x>:COUPling {AC|DC|GND|DCREJect}
CH<x>:COUPling?
```

Arguments `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.

Examples

`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.

Group

Vertical

Syntax

```
CH<x>:DESKew <NR3>
CH<x>:DESKew?
```

Arguments

`<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>: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 <str>
CH<x>:LABel:NAMe?
```

Arguments

`<str>` is an alphanumeric character string, ranging from 1 through 32 characters in length.

Examples

`CH2:LABel:NAMe "Pressure"` changes the waveform label for the CH2 waveform to "Pressure".

`CH3:LABel:NAMe?` might return `:CH3:LABEL:NAME "CH3"` indicating that the waveform label for the CH 3 waveform is "CH3".

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

Syntax

```
CH<x>:LABel:XPOS <NR1>
CH<x>:LABel:XPOS?
```

Arguments

`<NR1>` is the location (in pixels) 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 500.

Examples

`CH3:LABel:XPOS 50` moves the waveform label for the CH3 waveform so that it begins 50 pixels to the right of the left edge of the screen.

`CH2:LABel:XPOS?` might return `:CH2:LABEL:XPOS 50` indicating that the waveform label for the CH2 waveform is currently 50 pixels to the right of the left edge of the screen.

CH<x>:LABel: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 top 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 Y Pos.

Group

Vertical

Syntax

```
CH<x>:LABel:YPOS <NR1>
CH<x>:LABel:YPOS?
```

Arguments

`<NR1>` is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 385.

Examples

`CH3:LABel:YPOS -25` moves the waveform label for the CH3 waveform to just beneath (25 pixels below) the top of the screen.

`CH2:LABel:YPOS?` might return `:CH2:LABEL:YPOS 0` indicating that the waveform label for the CH2 waveform is currently located just beneath the top of the screen.

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, REF<x>:VERTical:POSition, MATH<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 autozeros 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:AUTOZero EXECute

Arguments EXECUTE causes the attached probe to autozero.

Examples CH1:PROBE:AUTOZEROautozeros the probe attached to Ch1.

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 Ch1 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 `CH<x>:PROBE:DEGAUSS:STATE?` might return `CH2:PROBE:DEGAUSS:STATE PASSED` indicating that the probe has been degaussed.

CH<x>:PRObe:FORCEDRange

This command sets the attached probe to its default range, or it queries the default 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.

Group Vertical

Syntax

```
CH<x>:PRObe:FORCEDRange
CH<x>:PRObe:FORCEDRange?
```

Examples `CH1:PRObe:FORCEDRange` sets the attached probe to its default range.

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:RANge

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

Group

Vertical

Syntax

```
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.

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 10.0000E+06` indicating that the input resistance of the probe attached to Channel 2 is 10 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 below:

`ATTENuation 5X` sets the P7260 probe to ±0.75 V dynamic range with 6 GHz bandwidth and 5X attenuation.

`ATTENuation 25X` sets the P7260 probe to ±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 ±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 ±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 ±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:SIGnal {PASS|BYPass}
CH<x>: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

`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 the relay is closed preventing your channel 1 signal from reaching the instrument.

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?

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.

Group Vertical

Syntax `CH<x>:PROBECal?`

Returns This query will return one of the following:

- `FAIl` signifies that the probe calibration has failed for the selected channel.
- `INIT` signifies that the probe calibration has not yet been run for the selected channel.
- `PASS` signifies the probe calibration has passed for the selected channel.

Examples `CH2:PROBECal?` might return `:CH2:PROBECAL PASS` indicating that the probe calibration has passed for channel 2.

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 -200.00 to 200.00 dB.

Examples CH3:PROBEFUNC:EXTDBATTEN 2.5 T 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 <str>
CH<x>:PROBEFunc:EXTUnits?
```

Related Commands CH<x>:PROBe:UNIts?

Arguments

<str> 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 as well as 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

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 Ω 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 or 1,000,000 Ω.

Examples `CH4:TERMINATION 50.0E+0` establishes 50 Ω impedance on channel 1.

`CH2:TERMINATION?` might return `:CH2:TERMINATION 50.0E+0` indicating that channel 2 is set to 50 Ω impedance.

*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

`CMDBATCH OFF` disables command batching.

`CMDBATCH ?` might return `:CMDBATCH 1` indicating that command batching is turned on.

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 :PRODDELTA
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:FUNCTION VBArs` 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}
CURSor:LINESTyle?
```

Arguments

SOLID displays the cursors as solid lines.

DASHed displays the cursors as dashed lines.

Examples

CURSOR:LINESTYLE SOLID 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:MODE TRACK 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.

Group

Cursor

Syntax

```
CURSor:SCREEN:STYle {LINE_X|LINES|X}
CURSor:SCREEN: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:SCREEN:STYLE LINES 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.

Group

Cursor

Syntax

```
CURSor:SCREEN:XPOSITION<x>
CURSor:SCREEN:XPOSITION<x>?
```

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>
CURSor:SCREEN:YPOSITION<x>?
```

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

Arguments

`CH<x>` sets an input channel waveform as the cursor source. The valid channel waveform range is from 1 through 4.

`MATH<x>` sets a math waveform as the cursor source. The valid math waveform range is from 1 through 4.

`REF<x>` 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:STATE ON displays the cursors.

CURSOR:STATE ? might return :CURSOR:STATE 1 indicating that the cursors are active on the display.

CURSor:VBArs

This command returns 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:VBARS SNAp 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

Syntax

```
CURSor:VBArs:POS<x>
CURSor:VBArs:POS<x>?
```

Related Commands

CURSor:VBArs:UNIts

Arguments

`<NR3>` specifies the cursor position.

Returns

A floating point value with an exponent.

Examples

`CURSor:VBArs:POS2 9.00E-6` positions the cursor2 vertical bar cursor at 9 ms.

`CURSor:VBArs:POS1?` this command might return `:CURSOR:VBARS:POS1 1.0000E-06` 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.

Group

Cursor

Syntax

```
CURSor:VBArs:UNIts {SECOnds|HERtz}
CURSor:VBArs:UNIts?
```

Arguments

`SECOnds` sets the units of the vertical bar cursors for the time domain (seconds).

`HERtz` sets the units of the vertical bar cursors for the frequency domain (Hertz).

Returns

A string for `SECOnds` or `HERtz`, depending on the current vertical bar cursor units.

Examples

`CURSOR:VBARS:UNITS HERtz` sets the units of the VBArs cursors to 1/seconds.

`CURSOR:VBARS:UNITS ?` might return `:CURSOR:VBARS:UNITS SECONDS` indicating that the units for the vertical bar cursor are currently set to seconds.

CURSor:WAVEform

This command returns the current settings for the waveform cursors or forces them to snap to positions specified by the DATA:START and DATA:STOP commands.

Group

Cursor

Syntax

```
CURSor:WAVEform SNAp
CURSor:WAVEform?
```

Related Commands

DATa:STARt, DATa:STOP

Arguments

`SNAp` forces the position of waveform cursor 1 and 2 to snap to the waveform record points specified by the DATA:START and DATA:STOP commands, respectively.

Examples

`CURSOR:WAVEFORM SNAp` 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.

Group Cursor

Syntax

```
CURSor:WAVEform:HDELTA?
```

Examples `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.

Group 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:POSition<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:POSition<x> {<NR3>}
CURSor:WAVEform:POSition<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:POSition2 9.00E-6` sets the waveform cursor 2 position to 9 µs.

`CURSor:WAVEform:POSition2?` might return
`:CURSOR:WAVEFORM:POSITION2 51.0701402806E-009`
indicating waveform cursor 2 is at 51.1 ns.

CURSor:WAVEform:SOUrce<x>

This command sets or queries the source for a waveform cursor. The cursor is specified by x, which can be 1 or 2.

Group

Cursor

Syntax

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

Arguments

`CH<x>` sets an input channel waveform as the cursor source. The valid channel waveform range is from 1 through 4.

`MATH<x>` sets a math waveform as the cursor source. The valid math waveform range is from 1 through 4.

`REF<x>` sets a reference waveform as the cursor source. The valid reference waveform range is from 1 through 4.

Examples

`CURSor:WAVEform:SOUrce CH2` sets the Cursor 1 source to channel 2.

`CURSor:WAVEform:SOUrce?` might return `:CURSOR:WAVEFORM:SOURCE1 CH2` indicating that the Cursor1 source is channel 2.

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:STYLE LINES 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:UNITS BASe 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:PRODDELTA 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:PRODDELta? (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 ΔX × ΔY value is calculated as (Cursor 2 X - Cursor 1 X) × (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:PRODDELta?`

Examples `CURSOR:XY:PRODDELTA?` might return `:CURSOR:XY:PRODDELTA -24.9856` indicating that the ΔX × ΔY 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.*

Group Cursor

Syntax

```
CURSor:XY:PRODUCT<x>?
```

Examples `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.*

Group Cursor

Syntax

```
CURSor:XY:RADIUS<x>?
```

Examples `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 (ΔY ¸ ΔX) when the CURSor:XY:READOUT is set to RATio. The ratio is calculated as (Cursor 2 Y - Cursor 1 Y) ÷ (Cursor 2 X - 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.*

Group

Cursor

Syntax

```
CURSor:XY:RATDELta?
```

Examples

`CURSOR:XY:RATDELTA?` might return `:CURSOR:XY:RATDELTA -1.0167` indicating that the ΔY/ΔX 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 ÷ 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.*

Group

Cursor

Syntax

```
CURSor:XY:RATIO<x>?
```

Examples

`CURSor:XY:RATIO1?` might return `:CURSOR:XY:RATIO1 15.8730E-3` 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 (Δr) value when the 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.*

Group

Cursor

Syntax

```
CURSor:XY:RDELta?
```

Examples

`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).

Group

Cursor

Syntax

```
CURSor:XY:READOUT {RECTangular|POLARCoord|PRODuct|RATio}
CURSor:XY:READOUT?
```

Related Commands

DISplay:FORMat

Arguments

`RECTangular` displays the following values: X1, X2, ΔX, Y1, Y2, ΔY, t1, t2, Δt.

`POLARCoord` displays the following values: r1, r2, Δr, q1, q2, Δq, t1, t2, Δt.

`PRODuct` displays the following values: X1 x Y1, X2 xY2, ΔX x ΔY, t1, t2, ΔDt.

`RATio` displays the following values: Y1 ¸ X1, Y2 ¸ X2, ΔY ¸ x ΔX, t1, t2, Δt.

Examples

`CURSOR:XY:READOUT RECTangular` sets the XY cursor readout to rectangular.

`CURSOR:XY:READOUT ?` might return `:CURSOR:XY:READOUT RECTANGULAR` indicating that the XY cursor readout is set to rectangular mode.

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.*

Group

Cursor

Syntax

```
CURSor:XY:RECTX<x> {<NR3>}
CURSor:XY:RECTX<x>?
```

Arguments

`<NR3>` is the X position of the specified cursor in rectangular coordinates.

Examples

`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.*

Group

Cursor

Syntax

```
CURSor:XY:RECTY<x> {<NR3>}
CURSor:XY:RECTY<x>?
```

Arguments

`<NR3>` is the Y position of the specified cursor in rectangular coordinates.

Examples `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 µV.

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 {<Block>|<asc curve>}
CURVe?
```

Related Commands

DATa:DESTination, DATa:SOUrce, DATa:STARt, DATa:STOP, SAVe:WAVEform, SAVe:WAVEform:FILEFormat, WFMInpre?, WFMInpre:BYT_Nr, WFMOutpre?, ,

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)
>
> <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, dependent 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. In order to 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 in order to ensure that all data is transferred and read.*

CURVENext? (Query Only)

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 :CURVENext?. 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 :CURVENext? response.

Group

Waveform Transfer

Syntax

```
CURVENext?
```

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.

Group

Waveform Transfer

Syntax

```
CURVEStream {<Block>|<asc curve>}
CURVEStream?
```

Related Commands

CURVe, DATa:DESTination, DATa:SOUrce, DATa:STARt, DATa:STOP, SAVe:WAVEform, SAVe:WAVEform:FILEFormat, WFMInpre?, WFMInpre:BYT_Nr, WFMOutpre?, ,

Arguments

See the CURVe command.

Returns

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

Syntax

```
DATa {INIT|SNAp}
DATa?
```

Related Commands CURVe, DATa:STARt, DATa:STOP, DATa:ENCdg, SAVe:WAVEform:FILEFormat, , WFMInpre:NR_Pt, , WFMOutpre:NR_Pt?

Arguments `INIT` initializes the waveform data parameters to their factory defaults.

`SNAp` sets DATa:STARt and DATa:STOP to match the current V Bar/Paired/Split cursor positions.

Examples `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.

Group Waveform Transfer

Syntax

```
DATa:DESTination REF<x>
DATa:DESTination?
```

Related Commands CURVe

Arguments `REF<x>` is the reference where the waveform will be stored. The reference number is specified by x, which ranges from 1 through 4.

Examples `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:ENGdg 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 × 10^{38} to 3.4 × 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 2-15: DATa and WFMOutpre Parameter Settings

DATa:ENCdg	WFMOutpre Settings			
Setting	: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:ENCDG RPBinary` 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: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:SOURCE CH1` specifies that the CH1 waveforms will be transferred in the next CURVe? query.

`DATA:SOURCE ?` might return `:DATA:SOURCE REF3` indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

DATa: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, WFMOutpre:NR_Pt?, ,

Arguments

<NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATa:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is greater than DATa:STARt, the values will be swapped internally for the CURVE? query.

Examples

DATA:START 10 specifies that the waveform transfer will begin with data point 10.

DATA:START ? might return :DATA:START 214 indicating that data point 214 is the first waveform data point that will be transferred.

DATa:STOP

This command sets or queries the last data point that will be transferred when using the CURVe? query. This command allows for the transfer of partial waveforms to the controller.

Changes to the record length value are not automatically reflected in the data:stop value. As record length is varied, the data:stop value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not automatically and correctly reflect increases in record length if the distance from data:start to data:stop stays smaller than the increased record length.

NOTE. *When using the CURVe command, DATa:STOP is ignored 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, WFMOutpre:NR_Pt?, ,

Arguments

`<NR1>` is the last data point that will be transferred, which ranges from 1 to the record length. If `<NR1>` is greater than the record length, then data will be transferred up to the record length. If both DATa:STARt and DATa:STOP are greater than the record length, the last data point in the record is returned.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is less than DATa:STARt, the values will be swapped internally for the CURVE? query.

If you always want to transfer complete waveforms, set DATa:STARt to 1 and DATa:STOP to the maximum record length, or larger.

Examples

`DATA:STOP 15000` specifies that the waveform transfer will stop at data point 15000.

`DATA:STOP ?` might return `:DATA:STOP 14900` indicating that 14900 is the last waveform data point that will be transferred.

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.

Group

Miscellaneous

Syntax

```
DATE <QString>
DATE?
```

Related Commands

TIME

Arguments

`<QString>` is a date in the form "yyyy-mm-dd" where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

Examples DATE "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, RECAll:SETUp, *RST, *SAV, SAVe:SETUp, *SDS

Arguments

ALL argument deletes all the stored setups.

<NR1> argument 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

RECAll: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

`DESE 209` sets the DESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.

`DESE ?` might return `:DESE 186`, showing that the DESER contains the binary value 10111010.

DIAg:CONTROL:HALT

This command determines or queries whether the next execution of diagnostics will stop on the first diagnostic failure that occurs or will execute 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 to halt at the first diagnostic failure that occurs.

`OFF` disables the halt function, allowing the instrument to execute the entire set of diagnostics before halting, 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 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 ON clears pass/fail information from data structures, not the Event Log, at the start of tests.

OFF does not clear pass/fail information at the start of tests.

<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.

Examples DIAG:FAILURES:CLEAR ON clears pass/fail information on the start of tests.

DIAG:FAILURES:CLEAR? might return DIAG:FAILURES:CLEAR 1 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 returns 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:LEVEL AREA 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.

Group

Diagnostics

Syntax

```
DIAg:RESults?
```

Related Commands

DIAg:RESults:VERBose?

Examples

`DIAG:RESULTS?` might return `:DIAG:RESULTS "135"` indicating the specific failure code of the first test failure that occurred.

`DIAG:RESULTS?` might return `:DIAG:RESULTS "Pass"` indicating that the last diagnostic test passed.

`DIAG:RESULTS?` might return `:DIAG:RESULTS "*****"` indicating that the diagnostic test was not run.

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.

Group

Diagnostics

Syntax

```
DIAg:RESults:VERBose?
```

Related Commands

DIAg:RESults?

Examples

`DIAG:RESULTS:VERBOSE?` might return

```
:DIAG:RESULTS:VERBOSE "DIAG ABORTED, ERROR 15 in A2D-Dmux
1 interconnects".
```

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.

Group

Diagnostics

Syntax

```
DIAg:SELect:ALL ALL
```

Related Commands

DIAg:NAMe:AREA?, DIAg:NAMe:SUBSYS?, DIAg:NAMe:TEST?, DIAg:EXECUTE

Arguments

`ALL` selects all available diagnostics.

Examples

`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.*

Group

Diagnostics

Syntax

```
DIAg:SELect:AREA <NR1>
DIAg:SELect:AREA?
```

Related Commands

DIAg:SELect:SUBSYS, DIAg:SELect:TEST, DIAg:SELect:LAST

Arguments

`<NR1>` selects a diagnostic area by number, which can range from 0 (zero selects all) through 15 (as specified by DIAg:NUMITEMS?).

Examples

`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).

Group

Diagnostics

Syntax

```
DIAg:STATE {EXECUTE|ON|OFF|<NR1>}
DIAg:STATE?
```

Related Commands

TEST:STOP

Arguments

`EXECUTE` starts execution of the diagnostics.

`ON` puts the instrument into the state in which diagnostics can be run. This argument is thrown automatically if the DIAg:EXECUTE command is executed.

`OFF` disables diagnostics capabilities and returns the instrument to a normal operating state.

`<NR1>` = 0 disables diagnostics capabilities and returns the instrument to a normal operating state; any other value enables diagnostics.

Examples

`DIAG:STATE OFF` turns off diagnostics capabilities and returns the instrument to a normal operating state.

`DIAG:STATE?` might return `:DIAG:STATE 0` indicating that diagnostics are disabled.

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.

Group

Diagnostics

Syntax

```
DIAg:STOP
```

Related Commands

TEST:STOP

Examples

`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:CLOCK ON 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.

Group

Display Control

Syntax

```
DISplay:COLOr:MATHCOLOr {DEFAULT|INHERIT}
DISplay:COLOr:MATHCOLOr?
```

Related Commands

DISplay:COLOr:PALEtte:IMAGEView, DISplay:COLOr:PALEtte:RECORDView

Arguments

`DEFAULT` sets color math traces in nominal palette math color, which is red.

`INHERIT` sets color math traces in the source waveform color. Math waveforms are drawn in the same color as their primary source waveform.

Examples

`DISPLAY:COLOR:MATHCOLOR DEFAULT` sets the color math traces in the nominal palette math color.

`DISPLAY:COLOR:MATHCOLOR ?` might return `:DISPLAY:COLOR:MATHCOLOR INHERIT` indicating that the primary source waveform color is used for each math trace.

DISplay:COLOr:PALEtte:IMAGEView

This command sets or queries the color palette for all image view (DPO and WfmDB) traces.

Group

Display Control

Syntax

```
DISplay:COLOr:PALEtte:IMAGEView
{NORMal|MONOGREEN|MONOGRAY|TEMPErature|SPECTral|USEr}
DISplay:COLOr:PALEtte:IMAGEView?
```

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`

All traces share a multicolored palette, where "intensity" is represented by hue; blue for least frequently hit, red for most frequently hit.

`SPECTral`

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:IMAGEVIEW MONOGRAY` 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|SPECTral|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.

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

USEr colors all traces using a user-defined palette. All traces share this palette.

Examples

DISPLAY:COLOR:PALETTE:RECORDVIEW MONOGRAY 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.

Group

Display Control

Syntax

```
DISplay:COLOr:PALEtte:USEr RESET
DISplay:COLOr:PALEtte:USEr?
```

Arguments

RESET sets all user palettes to their default values.

Examples

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.

Group Display Control

Syntax

```
DISplay:COLOr:PALEtte:USEr:CARet <NR1>,<NR1>,<NR1>
DISplay:COLOr:PALEtte:USEr:CARet?
```

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:CARET 330,75,100` 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?
```

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:GRATICULE 330,75,100` 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.

Group Display Control

Syntax

```
DISplay:COLOr:PALEtte:USEr:HIStogram <NR1>,<NR1>,<NR1>
DISplay:COLOr:PALEtte:USEr:HIStogram?
```

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:HISTOGRAM 330,75,100 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.

Group

Display Control

Syntax

```
DISplay:COLOr:PALEtte:USEr:MASK <NR1>,<NR1>,<NR1>
DISplay:COLOr:PALEtte:USEr:MASK?
```

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:MASK 330,75,100 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.

Group Display Control

Syntax

```
DISplay:COLOr:PALEtte:USEr:MASKHighlight <NR1>,<NR1>,<NR1>
DISplay:COLOr:PALEtte:USEr:MASKHighlight?
```

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:MASKHIGHLIGHT 330,75,100 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 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 waveform 1 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 waveform 2 is assigned a hue of 300, and lightness of 90, and a saturation of 100.

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

Syntax

```
DISplay:COLOr:REFCOLOr {DEFAULT|INHERIT}
DISplay:COLOr:REFCOLOr?
```

Related Commands

DISplay:INTENSITy:WAVEform:IMAGEView, DISplay:INTENSITy:WAVEform:RECORDView

Arguments

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.

Examples

DISPLAY:COLOR:REFCOLOR DEFAULT 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:DESKew

This command controls or queries the state of the Display Only button.

Group

Display Control

Syntax

```
DISplay:DESKew {<NR1>|OFF|ON}
DISplay:DESKew?
```

Arguments

OFF sets deskew for the acquisition waveform.

ON sets deskew to deskew the display only.

<NR1> = 0 will deskew the acquisition waveform; any other value will deskew the display only.

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.

Group

Display Control

Syntax

```
DISplay:FILTer {LINEAr|SINX}
DISplay:FILTer?
```

Arguments

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

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

Examples

DISPLAY:FILTER SINX 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.

Group

Display Control

Syntax

```
DISplay:FORMat {YT|XY}
DISplay:FORMat?
```

Arguments

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.

Table 2-16: XY Format Pairs

X-Axis Source	Y-Axis Source
CH 1	CH 3
CH 2	CH 4
Ref 1	Ref 2
Ref 3	Ref 4

Examples

DISPLAY:FORMAT YT 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|FULl|GRId|IRE|NTSC|MV|PAL}
DISplay:GRAticule?
```

Arguments

`CROSSHair` specifies a frame and cross hairs.

`FRAme` specifies a frame only.

`FULl` 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:GRATICULE FRAme` 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 <NR1>
DISplay:INTENSITy:WAVEform:IMAGEView?
```

Related Commands DISplay:INTENSITy:WAVEform:RECORDView,

Arguments `<NR1>` is the waveform saturation and ranges from 10 to 100 percent.

Examples `DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW 100` 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 <NR1>
DISplay:INTENSITy:WAVEform:RECORDView?
```

Related Commands DISplay:INTENSITy:WAVEform:IMAGEView,

Arguments `<NR1>` is the waveform saturation and ranges from 10 to 100 percent.

Examples

`DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW 100` 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:PERSISTENCE VARPersist` sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARpersist command.

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

DISplay:PERSistence:RESET (No Query Form)

This command controls the clearing of persistence data.

Group

Display Control

Syntax

```
DISplay:PERSistence:RESET
```

Related Commands DISplay:PERSistence, DISplay:VARpersist

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

DISplay:SCREENText? (Query Only)

This command returns all screen text settings.

Group Display Control

Syntax

```
DISplay:SCREENText?
```

Related Commands DISplay:SCREENText:STATE, DISplay:SCREENText:LABel<x>?, DISplay:SCREENText:LABel<x>:NAMe, DISplay:SCREENText:LABel<x>:XPOS, DISplay:SCREENText:LABel<x>:YPOS

Examples `DISPLAY:SCREENTEXT?` might reply:

```
:DISPLAY:SCREENTEXT:STATE
0;LABEL1:NAME "";XPOS 100;YPOS
5;:DISPLAY:SCREENTEXT:LABEL2:NAME "";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
65;: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:SCREENText:LABel<x>? (Query Only)

This 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:SCREENText:LABel<x>?
```

Related Commands DISplay:SCREENTExt?, DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>:NAMe, DISplay:SCREENTExt:LABel<x>:XPOS, DISplay:SCREENTExt:LABel<x>:YPOS

Examples `DISPLAY:SCREENTEXT:LABEL3?` might return `:DISPLAY:SCREENTEXT: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: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>
```

Related Commands , DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>:XPOS, DISplay:SCREENTExt:LABel<x>:YPOS

Arguments `<QString>` argument is the text to be displayed for a given label.

Examples `DISPLAY:SCREENTEXT:LABEL2:NAME "THIS IS SCREEN TEXT"` sets the screen text for label 2 to THIS IS SCREEN TEXT.

`DISplay:SCREENTExt:LABel2:NAMe?` might return `:DISPLAY:SCREENTEXT:LABEL2:NAME "MyWaveform"` indicating that the label 2 screen text is set to MyWaveform.

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.

Group Display Control

Syntax

```
DISplay:SCREENTExt:LABel<x>:XPOS <NR1>
```

Related Commands DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>:NAMe, DISplay:SCREENTExt:LABel<x>:YPOS

Arguments `<NR1>` argument sets the horizontal position of the screen text label. The value can range from 0 to 500.

Examples `DISPLAY:SCREENTEXT:LABEL3:XPOS 100` sets the horizontal position of label 3 to 100.

`DISPLAY:SCREENTEXT:LABEL3:XPOS?` might return. `:DISPLAY:SCREENTEXT:LABEL3:XPOS 250` indicating that screen label 3 is horizontally positioned to 250.

DISplay:SCREENTExt: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.

Group Display Control

Syntax

```
DISplay:SCREENTExt:LABel<x>:YPOS <NR1>
```

Related Commands DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>:NAMe, DISplay:SCREENTExt:LABel<x>:XPOS

Arguments `<NR1>` argument sets the vertical position of the screen text label. The value can range from 0 to 385.

Examples `DISPLAY:SCREENTEXT:LABEL3:YPOS 100` sets the vertical position of label 3 to 100.

`DISPLAY:SCREENTEXT:LABEL3:YPOS?` might return `:DISPLAY:SCREENTEXT:LABEL3:YPOS 192` indicating that screen label 3 is vertically positioned to 192.

DISplay:SCREENTExt:STATE

This command controls the display of screen text.

Group 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:STATE ON` 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 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:SHOWREMOTE ON` 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

Syntax

```
DISplay:TRIGBar {OFF|SHORt|LONG}
DISplay:TRIGBar?
```

Arguments

OFF removes the trigger indicator bar from the display.

SHORt displays, as the indicator, a short arrow at the right side of the graticule for each displayed trigger signal.

LONG displays, as the indicator, a horizontal line across the width of the graticule for each displayed trigger signal.

Examples

DISPLAY:TRIGBAR LONG sets the display to show a long trigger indicator bar (or bars).

DISPLAY:TRIGBAR? might return :DISPLAY:TRIGBAR OFF, indicating that the indicator bar is removed from the display.

DISplay:TRIGT

This command controls or queries the display of the trigger T. The trigger T shows where the trigger occurred on the waveform.

Group

Display Control

Syntax

```
DISplay:TRIGT {<NR1>|OFF|ON}
DISplay:TRIGT?
```

Arguments

OFF removes the trigger indicator T from the display.

ON displays a T at the trigger point.

<NR1> = 0 disables the trigger T; any other value displays the trigger T.

Examples

DISPLAY:TRIGT OFF disables the trigger T.

DISPLAY:TRIGT ? might return :TRIGT 1 indicating that the trigger T is displayed.

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.

Group

Display Control

Syntax

```
DISplay:VARpersist <NR3>
DISplay:VARpersist?
```

Related Commands

DISplay:PERSistence

Arguments

`<NR3>` indicates the persistence decay time setting in numeric seconds. The persistence time ranges from 50 ms to 100 s.

Examples

`DISPLAY:VARPERSIST 5E-1` sets the persistence decay time to 500 ms.

`DISPLAY:VARPERSIST?` might return `:DISPLAY:VARPERSIST 5.0000E-01` indicating that persistence decay time is currently set to 500 ms.

DISplay:WAVEform

This command controls or queries the display of acquisition and reference waveforms.

Group

Display Control

Syntax

```
DISplay:WAVEform {<NR1>|OFF|ON}
DISplay:WAVEform?
```

Arguments

`OFF` removes the waveforms from the display.

`ON` displays waveforms on the display.

`<NR1>` = 0 removes waveforms from the display; any other value displays the waveforms.

Examples

`DISPLAY:WAVEFORM OFF` removes waveforms from the display.

`DISPLAY:WAVEFORM?` might return `:WAVEFORM 1` indicating that waveforms are displayed.

EMail (No Query Form)

This command (no query form) sends a test e-mail message or sets the current e-mail sent count to zero.

Group

Email

Syntax

```
EMail {TESt|RESET}
```

Arguments

`TESt` argument sends a test e-mail message.

`RESET` argument sets the e-mail sent count to zero.

Examples

`EMAIL TEST`

This command sends a test e-mail.

`EMAIL RESET` resets the sent mail count to zero.

EMail: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.

Group

Email

Syntax

```
EMail:ATTempts <NR1>
EMail:ATTempts?
```

Arguments

`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.

Examples

`EMAIL:ATTEMPTS 15` 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.

Group

Email

Syntax

```
EMail:AUTHLogin {<QString>}
EMail:AUTHLogin?
```

Arguments

<QString> argument is a string that specifies the login name to be used with the SMTP e-mail server.

Examples

EMAIL:AUTHLOGIN kentb 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.

Group

Email

Syntax

```
EMail:AUTHPassword {<QString>}
```

Arguments

<QString> argument is a string that specifies the password to be used with the SMTP e-mail server.

Examples

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.

Group

Email

Syntax

```
EMail:COUNt?
```

Examples

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.

Group

Email

Syntax

```
EMail:FROm {<QString>}
EMail:FROm?
```

Arguments

<QString> argument is a string that is placed in the From line of the e-mail. An example is "johnz".

Examples

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 hostname that will be used when e-mail is sent to the SMTP e-mail server if the DPO hostname will not work.

Group

Email

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 hostname when sending e-mail to the SMTP e-mail server.

EMAIL:HOSTWANTED ? might return :EMAIL:HOSTWANTED "LazyEight" indicating that LazyEight is the hostname 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

Email

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:IMAGE 1` 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

Email

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:MASK 1` 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 Email

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:MAXSIZE 5` 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 Email

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:MEASUREMENT 1` 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

Email

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:NUMEMAILS 10` 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

Email

Syntax

```
EMail:SMTPPort {<NR1>}
EMail:SMTPPort?
```

Arguments

`<NR1>` can be 1 through 65535. This number specifies the TCPIP port number.

Examples

`EMAIL:SMTPPORT 55` 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 Email

Syntax

```
EMail:SMTPServer <string>
EMail:SMTPServer?
```

Arguments `<string>` 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 Email

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 Email

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:TIMEOUT 60` 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.

EMail:TO

This command sets or queries the address of the recipient(s) of an e-mail.

Group Email

Syntax

```
EMail:TO <string>
EMail:TO?
```

Arguments

<string> 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 "billsmill@tek.com"` indicating that the billsmith@tek.com will receive e-mail when the event occurs.

EMail: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

Email

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:TRIGGER 1` 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

Email

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:WAVEFORM 1` 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 `*ESE 209` sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

`*ESE ?` might return 186, showing that the ESER contains 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 along 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 may 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 Hardcopy

Syntax

```
EXPort STARt
EXPort?
```

Arguments `STARt` initiates the export.

Examples `EXPORT START` 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\Images\hcFILE.BMP"`.

EXPort:FILEName

This command sets or returns the file/path that will be sent export data on the next EXPort command.

Group Hardcopy

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\Images". 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\Images\TEK.BMP"` as the selected file.

EXPort:FORMat

This command sets or returns the image format for exporting waveforms to a file.

Group Hardcopy

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:FORMAT BMP` sets the export format to BMP.

`EXPORT:FORMAT ?` might return `:EXPORT:FORMAT JPEG` indicating that waveforms will be exported in the JPEG format.

EXPort:IMAGe

This command sets or returns the inksaver mode for exporting waveforms (File > Export Setup > Images > Image).

Group Hardcopy

Syntax

```
EXPort:IMAGE {NORMal|INKSaver|ENHANCedwfm}
EXPort:IMAGe?
```

Arguments

`NORMal` preserves the screen background when printing waveforms.

`INKSaver` changes the screen background to white. This saves ink when printing waveforms.

`ENHANCedwfm` prints (color) screen images with colors designed to print on a white background.

Examples

`EXPORT:IMAGE NORMAL` sets the export mode to Normal.

`EXPORT:IMAGE ?` might return `:EXPORT:IMAGE NORMAL` indicating that the screen background will be printed when printing waveforms.

EXPort:PALEtte

This command sets or returns the export color palette.

Group

Hardcopy

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:READOuts

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

Hardcopy

Syntax

```
EXPort:READOuts {OFFGRAticule|ONGRAticule}
EXPort:READOuts?
```

Arguments

`OFFGRATICULE` places the readouts off the graticule area.

`ONGRATICULE` places the readouts on the graticule area for export.

EXPort:VIEW

This command sets or returns the area of the screen to be exported.

Group

Hardcopy

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:VIEW FULLSCREEN 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.
- Enables all Command Headers.
- 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: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:STATE ON 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 `<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\images"` sets the current working directory to images.

`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>, GPIb
```

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.

`GPIb` specifies that the hard copy is sent out of the GPIB port. GPIb is the only port allowed for this command.

Examples

`FILESYSTEM:READFILE "C:\TEK00000.IBM",GPIb` 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

`HARDCOPY START` 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` "C:\TekScope\Images\hcFILE.BMP";PORT FILE.

HARDCopy:FILEName

This command sets or returns 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\Images". 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\Images\TEK.BMP"` as the selected file.

`HARDCOPY:FILENAME "TEK.BMP"` sets TEK as the selected file name with a numeric, two-digit suffix. The instrument might return `:HARDCOPY:FILENAME "TEK00.BMP";"TEK01.BMP"` This indicates TEK00.BMP is the first file and TEK01.BMP is the second file.

HARDCopy:IMAGe

This command sets or returns the inksaver mode for hard copy printing.

Group

Hardcopy

Syntax

```
HARDCopy:IMAGe {NORMal|INKSaver|ENHANCedwfm}
HARDCopy:IMAGe?
```

Arguments

NORMal preserves the screen background when printing waveforms.

INKSaver changes the screen background to white. This saves ink when printing waveforms.

ENHANCedwfmprints screen images with colors designed to print on a white background.

Examples

HARDCOPY:IMAGE NORMAL sets the hard copy mode to Normal.

HARDCOPY:IMAGE ? might return :HARDCOPY:IMAGE INKSAVER indicating that the inksaver mode is on and the screen background is white.

HARDCopy:LAYout

This command sets or returns 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:LAYOUT LANDSCAPE 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 returns 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:PALETTE COLOR` 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.

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:PORT FILE` 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.

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 returns 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:VIEW FULLSCREEN` 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

`HDR OFF` 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 `HEADER OFF` specifies that the instrument omits headers on query responses, so that only the argument is returned.

`HEADER ?` might return `:HEADER 1` indicating that the instrument is including headers on applicable query responses.

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:BOXPcnt 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:BOXPcnt

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:BOX 1E-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 <NR3>, <NR3>, <NR3>, <NR3>
HIStogram:BOXPcnt?
```

Related Commands HIStogram:BOX

Arguments `<NR3>` specifies the left position of the histogram box in percentage coordinates. The default value is 30.0.

`<NR3>` specifies the top position of the histogram box in percentage coordinates. The default value is 25.1.

`<NR3>` specifies the right position of the histogram box in percentage coordinates. The default value is 70.0.

`<NR3>` specifies the bottom position of the histogram box in percentage coordinates. The default value is 75.2.

Examples `HISTOGRAM:BOXPCNT 30.0,25.0,70.0,75.0` 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 × 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 × 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:BOXPcnt

Examples `HISTOGRAM:COUNT RESET` 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".*

The Histogram:DATa? query corresponds to the Histogram Data (CSV) controls included in the instrument user interface. To find these controls, select either Edit > Copy > Setup > Measurements tab or the File > Save As > Histogram button.

This command returns only histogram data values separated by commas on a single line. The CSV file created by the user interface actions indicated above, writes out a line for each histogram data point, and contains the time relative to the trigger, a comma, and the histogram value.

Group Histogram

Syntax

```
HIStogram:DATa?
```

Related Commands HIStogram:BOX, HIStogram:BOXPcnt, HIStogram:COUNt, HIStogram:DISplay, HIStogram:FUNCtion, HIStogram:MODe, HIStogram:SIZe, HIStogram:SOUrce, HIStogram:STATE

Examples `HISTOGRAM:DATA?` might return

```
:HISTOGRAM:DATA
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,1245,933,903,1003,838,782,764,697,
687,620,582,566,530,595,557,552,518,524,606,
503,526,470,498,554,534,487,546,548,504,554,
513,566,639,508,643,704,662,769,785,806,922,
960,1153,1336,1478,1960,4515,17598,52184,
67040,50417,12885,4104,2382,1866,1699,1428,
1288,1271,1140,975,1024,894,841,841,678,841,
804,685,713,746,729,644,640,787,768,668,671,
661,745,755,692,882,772,802,862,770,854,799,
969,1102,1080,1040,1272,1329,1394,1760,1428,
2418,3313,4475,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0.
```

HIStogram:DISplay

This command sets the scaling of the histogram data display to be the count of each histogram bin or the log of that count. The default scaling is Linear.

The query form of this command returns either the histogram display scaling or the display status "off."

This command is equivalent to selecting Waveform Histogram from the Measure menu, and then selecting Linear or Log in the Histogram Scaling group.

Group

Histogram

Syntax

```
HIStogram:DISplay {OFF|LOG|LINEAr}
HIStogram:DISplay?
```

Related Commands

HIStogram:MODe, HIStogram:STATE

Arguments

`<OFF>` 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.

`<LOG>` sets the histogram display to logarithmic scaling.

`<LINEAr>` sets the histogram display to linear scaling. This is the default setting.

Examples

`HISTOGRAM:DISPLAY LOG` sets the histogram scaling to be the log count of each histogram bin.

`HISTOGRAM:DISPLAY ?` might return `:HISTOGRAM:DISPLAY LOG` indicating that the current histogram display scaling is the log count of each histogram bin.

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.

Group

Histogram

Syntax

```
HIStogram:FUNCtion {HORizontal|VERTical}
HIStogram:FUNCtion?
```

Related Commands

HIStogram:MODe, HIStogram:STATE

Arguments

`HORizontal` displays a horizontally positioned histogram that shows time distribution.

`VERTical` displays a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).

Examples

`HISTOGRAM:FUNCTION HORIZONTAL` 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:MODE HORIZONTAL` 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.

Group Histogram

Syntax

```
HIStogram:SIZe <NR3>
HIStogram:SIZe?
```

Related Commands HIStogram:DISplay

Arguments `<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.

Examples `HISTOGRAM:SIZE 2.0` 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.

Group

Histogram

Syntax

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

Related Commands

HIStogram:DISplay

Arguments

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:SOURCE CH1 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:STATE ON` 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 0.00000000000;:HORIZONTAL:MAIN:INTERPRATIO 1.0000;SCALE 40.0000E-9;POSITION 50.0000;SAMPLERATE 1.2500E+9;UNITS:STRING "s";:HORIZONTAL:RECORDLENGTH 500;RESOLUTION 500;ROLL AUTO;READOUTS 0;FASTFRAME:STATE 0`.

HORizontal:ACQDURATION? (Query Only)

This query returns the timebase duration.

Group

Horizontal

Syntax

```
HORizontal:ACQDURATION?
```

Returns

<NR3>

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>

Examples `HORIZONTAL:ACQLENGTH?` might return ::HORIZONTAL:ACQLENGTH 50000 indicating the acquisition record length is 50000 points.

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?`

Arguments None.

Examples `HORIZONTAL:DIVISIONS?` might return `:HORIZONTAL:DIVISIONS 10.0000` indicating that the waveform is displayed across ten division.

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

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 ≈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: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 µs.

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 query HORizontal:MAIn:SAMPLERate? returns the sampling rate (real and interpolated). The real maximum sampling rate, as specified by the acquisition hardware, is:

Max sampling rate =

HORizontal:MAIn;SAMPLERate?/HORizontal:MAIn:INTERPRatio?

The default value 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 ≈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:SAMPLERate

This command sets the horizontal sample rate to the desired number of samples per second. The query form of this command returns the current horizontal sample rate. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu and then entering the desired Sample Rate.

Group

Horizontal

Syntax

```
HORizontal:MAIn:SAMPLERate <NR3>
HORizontal:MAIn:SAMPLERate?
```

Related Commands

HORizontal:RESOlution

Arguments

`<NR3>` represents the size of the sample rate, which ranges from 5 S/s to 250 GS/s.

Examples

`HORIZONTAL:MAIN:SAMPLERATE 125E6` sets the sample rate to the sample rate increment that is closest to 125 MS/s.

`HORIZONTAL:MAIN:SAMPLERATE ?` might return `:HORIZONTAL:MAIN:SAMPLERATE 2.5000E+09` indicating that the sample rate is currently set to 2.5 GS/s.

HORizontal[:MAIn]:SCAle

This command sets the time per division for the time base or returns its horizontal scale on the display and is identical to the HORizontal:SCAle command. The specified scale value is rounded to a valid scale setting. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then choosing a Scale value.

Group

Horizontal

Syntax

```
HORizontal[:MAIn]:SCAle
HORizontal[:MAIn]:SCAle?
```

Related Commands

Arguments

`<NR3>` is the time per division. The range is from 200 ps through 40 s.

Examples

`HORIZONTAL[:MAIN]:SCALE 2E-6` sets the main scale to 2µs per division.

`HORIZONTAL[:MAIN]:SCALE ?` might return `:HORIZONTAL:MAIN:SCALE 2.0000E-06` indicating that the main scale is currently set to 2 µs per division.

HORizontal:MAIn:UNIts

This command sets or returns the units for the horizontal time base. It is equivalent to setting the HORizontal:MAIn:UNIts:STRing.

Group

Horizontal

Syntax

```
HORizontal:MAIn:UNIts <STRing>
HORizontal:MAIn:UNIts?
```

Related Commands

HORizontal:MAIn:UNIts:STRing

Arguments

`<STRing>` 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 returns the units string for the horizontal time base.

Group

Horizontal

Syntax

```
HORizontal:MAIn:UNIts:STRing <string>
HORizontal:MAIn:UNIts:STRing?
```

Related Commands

HORizontal:MAIn:UNIts

Arguments

`<string>` 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:RECOrdlength

This command sets the horizontal record length to the number of data points in each frame. The query form of this command returns the current horizontal record length. This command is equivalent to selecting Resolution from the Horiz/Acq menu and then entering the desired Rec Length.

Group

Horizontal

Syntax

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

Arguments

`<NR1>` represents the supported values for horizontal record lengths. For information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

Examples

`HORIZONTAL:RECORDLENGTH 50000` specifies that 50000 data points will be acquired for each record.

`HORIZONTAL:RECORDLENGTH` ? might return `:HORIZONTAL:RECOrdlength 5000` indicating that the horizontal record length is equal to 5000 data points.

HORizontal:RESOlution

This command sets the horizontal record length to the number of data points in each frame. The sample rate is automatically adjusted at the same time to maintain a constant time per division. The query form of this command returns the current horizontal record length. This command is equivalent to adjusting the RESOLUTION knob on the front panel.

Group

Horizontal

Syntax

```
HORizontal:RESOlution <NR1>
HORizontal:RESOlution?
```

Arguments <NR1> represents the supported values for horizontal record lengths. For information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

Examples HORIZONTAL:RESOLUTION 50000 specifies that 50000 data points will be acquired for each record.

HORIZONTAL:RESOLUTION ? might return :HORIZONTAL:RESOLUTION 5000 indicating that the horizontal record length is equal to 5000 data points.

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:ROLL ON 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>?
```

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>?`

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:01.00.912. This indicates the instrument model number, configured format, and firmware version number.

*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.

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

`LOCK ALL` 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.

MASK? (Query Only)

This query-only command returns the states of all settable mask parameters.

Group

Mask

Syntax

```
MASK?
```

Examples

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: 3A 4D 41 53 4B 3A 46 49 4C 54 45 52 20 30 3B 41 55 54 4F 41 44 4A 55 53 54 20 30 3B 41 55 54 4F 41 44 4A 55 53 54 3A 48 44 45 4C 54 41 20 31 30 3B 56 44 45 4C 54 41 20 31 30 3B 3A 4D 41 53 4B 3A 41 55 54 4F 53 45 54 3A 41 55 54 4F 41 44 4A 55 53 54 20 31 3B 48 50 4F 53 20 31 3B 48 53 43 41 4C 45 20 31 3B 4D 4F 44 45 20 4D 41 4E 55 41 4C 3B 4F 46 46 53 45 54 41 44 4A 20 31 3B 54 52 49 47 47 45 52 20 31 3B 56 50 4F 53 20 31 3B 56 53 43 41 4C 45 20 31 3B 53 54 41 4E 44 41 52 44 20 4E 4F 4E 45 3B 55 53 45 52 3A 4F 4E 45 20 30 2E 30 30 30 30 3B 5A 45 52 4F 20 30 2E 30 30 30 30 3B 54 59 50 45 20 4E 4F 52 4D 41 4C 49 5A 45 44 3B 3A 4D 41 53 4B 3A 53 4F 55 52 43 45 20 43 48 31 3B 43 4F

```
55 4E 54 3A 46 41 49 4C 55 52 45 53 20 30 3B 48 49 54 53
20 30 3B 53 45 47 31 3A mask?  :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;.
```

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.

Group

Mask

Syntax

```
MASK:AUTOAdjust {ON|OFF|<NR1>}
```

Arguments

ON enables the autoadjust function.

OFF disables the autoadjust function

<NR1> = 0 disables the autoadjust function; any other value enables it.

Examples

MASK:AUTOADJUST ON enables the autoadjust function.

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 persisted data. If Autofit makes frequent adjustments, there can be very little or even no persisted data displayed.

Group

Mask

Syntax

```
MASK:AUTOAdjust:HDELTA {<NR3>}
```

Arguments

<NR3>

Is a floating point number which represents a percent of a division.

Examples `MASK:AUTOADJUST:HDELTA 0.6` sets the horizontal autofit search to 0.6 division.

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 persisted data. If Autofit makes frequent adjustments, there can be very little or even no persisted data displayed.

Group Mask

Syntax

```
MASK:AUTOAdjust:VDELTA {<NR3>}
MASK:AUTOAdjust:VDELTA?
```

Arguments `<NR3>` is a floating point number which represents a percent of a division.

Examples `MASK:AUTOADJUST:VDELTA 0.6` sets the vertical autofit search to 0.6 of a division.

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.

Group Mask

Syntax

```
MASK:AUTOSet:AUTOAdjust {ON|OFF|<NR1>}
MASK:AUTOSet:AUTOAdjust?
```

Arguments `ON` enables the autoset:autoadjust function.

`OFF` disables the autoset:autoadjust function.

`<NR1>` = 0 disables the autoset:autoadjust function; any other value enables it.

Examples `MASK:AUTOSET:AUTOADJUST ON` enables the autoset autoadjust function.

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.

Group Mask

Syntax

```
MASK:AUTOSet:HPOS {ON|OFF|<NR1>}
```

Arguments `ON` enables the autoset:hpos function, causing ¼ to occur.

`OFF` disables the autoset:hpos function, causing ¼ to occur.

`<NR1>` = 0 disables the autoset:hpos function; any other value disables enables it.

Examples `MASK:AUTOSET:HPOS ON` enables the autoset:hpos function.

MASK:AUTOSet:HSCAle

This command controls whether the autoset algorithm will attempt to change the horizontal scale while attempting a mask autoset.

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:HSCAle {ON|OFF|<NR1>}
```

Arguments `ON` enables the autoset:hscale function.

`OFF` disables the autoset:hscale function.

`<NR1>` = 0 disables the autoset:hscale function; any other value enables it.

Examples `MASK:AUTOSET:HSCALE ON` enables the autoset:hscale function.

MASK:AUTOSet:MODe

This command controls whether or not a mask autoset will be automatically done after a standard mask is selected. The autoset will never happen if the standard mask is selected from GPIB, since a "SET?" sent back to the scope should not do a mask autoset.

The default value is: manual.

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:MODe {MANual|AUTO}
```

Arguments

`MANual` disables a mask autoset from occurring automatically after a standard mask is selected.

`AUTO` enables a mask autoset to automatically occur after a standard mask is selected.

Examples

`MASK:AUTOSET:MODE AUTO` enables a mask autoset to occur automatically after a standard mask is selected.

MASK:AUTOSet:OFFSETAdj

This command sets mask autoset to not enforce the rule that, for pulse standards, require 0V to be in a certain place in the mask. Instead, mask 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.

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:OFFSETAdj {ON|OFF|<NR1>}
MASK:AUTOSet:OFFSETAdj?
```

Arguments

`ON` enables the autoset:offsetadj function.

`OFF` disables the autoset:offsetadj function.

`<NR1>` = 0 disables the autoset:offsetadj function; any other value enables it.

Examples

`MASK:AUTOSET:OFFSETADJ ON` enables the autoset:offsetadj function.

MASK:AUTOSet:STANdard

This command selects the standard mask in a mask autoset.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general instrument autoset 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|ENETXAUI_Near
|ENETXAUI_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 (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Electrical 132.8 Mb/s)

FC266E (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Electrical, 132.8 Mb/s)

FC531E (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Electrical, 531.2 Mb/s)

FC1063E (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)

FC1063E_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC1063E_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC1063E_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC1063E_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC1063E_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC1063E_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC1063E_AGR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)

FC2125E_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC2125E_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC2125E_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC2125E_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC2125E_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)

FC2125E_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC2125E_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC2125E_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC2125E_AGR (ANS1 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)

`TFI5_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:STANDARD E1COAX` 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 controls if a mask AUTOSet affects the trigger level. Other trigger settings such as type of trigger are not changed by 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:TRIGger {ON|OFF|<NR1>}
```

Arguments

`ON` enables the autoset:trigger function.

`OFF` disables the autoset:trigger function.

`<NR1>` = 0 disables the autoset:trigger function; any other value enables it.

Examples

`MASK:AUTOSET:TRIGGER ON` enables the autoset:trigger function.

MASK:AUTOSet:USER:ONE

This command sets or queries the level 1 used by autoset 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 autoset for user masks.

Examples

`MASK:AUTOSET:USER:ONE?` might return :`MASK:AUTOSET:USER:ONE` indicating that autoset is using 0.0 volts for 1 in user masks.

MASK:AUTOSet:USER:TYPe

This command sets or queries how autoset 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 autoset 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? might return :MASK:AUTOSET:USER:TYPE NORMALIZED indicating that 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:ZERO? might return :MASK:AUTOSET:USER:ZERO 0.0000 indicating 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>}
```

Arguments

`ON` enables the autoset:vpos function.

`OFF` disables the autoset:vpos function.

`<NR1>` = 0 disables the autoset:vpos function; any other value enables it.

Examples

`MASK:AUTOSET:VPOS ON` enables the autoset:vpos function.

MASK:AUTOSet:VSCAle

This command controls whether the mask autoset algorithm will affect the vertical scale while attempting to autoset.

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:VSCAle {ON|OFF|<NR1>}
```

Arguments

`ON` enables the autoset:vscale function, causing ¼ to occur.

`OFF` disables the autoset:vscale function, causing ¼ to occur.

`<NR1>` = 0 disables the autoset:vscale function; any other value enables it.

Examples

`MASK:AUTOSET:VSCALE ON` enables the autoset:vscale function.

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.

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 command returns the number of pass/fail tests that have failed. The default is 0. To work 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 no tests have failed.

MASK:COUNt:HITS? (Query Only)

This command returns the number of hits on all mask segments. To work, hit counting must be turned on.

Group Mask

Syntax `MASK:COUNt:HITS?`

Returns <NR1>

Examples `MASK:COUNT:HITS?` might return `:MASK:COUNT:HITS 0` showing there were no hits from any source on all mask segments.

MASK:COUNt:SEG<m>:HITS? (Query Only)

This command returns the number of hits in mask segment <m>. To work, hit counting must be turned on.

Group Mask

Syntax `MASK:COUNt:SEG<m>:HITS?`

Arguments `<m>` is the mask segment number.

Returns <NR1>

Examples `MASK:COUNT:SEG1:HITS?` might return `:MASK:COUNT:SEG1:HITS 0` showing there have been no hits.

MASK:COUNt:STATE

Sets or returns the mask hits count state. In other words, it controls whether or not mask counting is being done. MASK:DISplay must be ON to enable MASK:COUNt:STATE to count mask violations.

Group Mask

Syntax `MASK:COUNt:STATE {OFF|ON|<NR1>}`

Arguments `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:STATE ON turns on mask hit counting.

MASK:COUNt:TESTS? (Query Only)

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)

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 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 50 test violations have occurred.

MASK:COUNt:WAVEFORMS? (Query Only)

This 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 1,000 waveforms have been acquired and processed.

MASK:DISplay

Sets or returns 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 not available 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:DISPLAY ON sets the display to show the defined masks

MASK:DISPLAY ? might return :MASK:DISPLAY 1, indicating that the display shows masks.

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>}
```

Related Commands

ACQuire:MODe

Arguments

ON enables the digital filter.

OFF disables the digital filter.

<NR1> = 0 disables the digital filter; any other value enables it.

Examples

MASK:FILTER ON enables the digital filter.

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

ON enables the mask:highlighthits function.

OFF disables the mask:highlighthits function.

<NR1> = 0 disables the mask:highlighthits function; any other value enables it.

Examples

MASK:HIGHLIGHTHITS ON enables the mask highlighthits function.

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>}
```

Arguments

ON

Negative

OFF

Positive

`<NR1>` = 0 disables the mask:invert function; any other value enables it.

Examples `MASK:INVERT ON` enables the mask invert function.

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.

Group Mask

Syntax

```
MASK:LOCk {OFF|ON|<NR1>}
MASK:LOCk?
```

Arguments `ON`

turns on mask lock to waveform.

`OFF`

turns off mask lock to waveform.

`<NR1>` = 0 turns off mask lock to waveform and any other value turns on mask lock to waveform.

Examples `MASK:LOCK ON` turns on mask lock to waveform.

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.

Group Mask

Syntax

```
MASK:MARgin:PERCent <NR3>
MASK:MARgin:PERCent?
```

Arguments

<NR3> values range from -50.0 to +50.0. The default is 5.

Examples

MASK:MARGIN:PERCENT 25 sets the mask margin percentage to 25.

MASK:MARGIN:PERCENT ? might return :MASK:MARGIN:PERCENT 1.00000000000E+001, indicating that the mask margin is set to 10%.

MASK:MARgin:STATE

This command sets or returns the state of the mask margins.

Group

Mask

Syntax

```
MASK:MARgin:STATE {OFF|ON|<NR1>}
```

Arguments

ON

Turns on mask margins.

OFF

Turns off mask margins. The currently displayed margined mask is erased and the original mask is displayed.

<NR1> = 0 turns off mask margins; any other integer turns on the selected mask margins.

Examples

MASK:MARGIN:STATE ON turns on mask margins

MASK:MARGIN:STATE ? might return :MASK:MARGIN:STATE 1 indicating that 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.

Group

Mask

Syntax

```
MASK:MASKPRE:AMPlitude <NR3>
MASK:MASKPRE:AMPlitude?
```

Related Commands MASK:MASKPRE:VSCAle

Arguments `<NR3>` is a floating number that sets the nominal pulse amplitude in volts

Examples `MASK:MASKPRE:AMPLITUDE 2.4` sets the current mask nominal pulse amplitude to 2.4 volts.

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.

Group Mask

Syntax

```
MASK:MASKPRE:HSCAle <NR3>
MASK:MASKPRE:HSCAle?
```

Related Commands MASK:MASKPRE:HTRIGPOS

Arguments `<NR3>` is a floating point number that sets the mask drawing timing resolution.

Examples `MASK:MASKPRE:HSCALE 40E-9` sets the nominal timing resolution used to draw the mask pulse shape to 40 nanoseconds per division.

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:HTRIGPOS 0.6` sets the current mask nominal pulse leading edge position to the sixth horizontal graticule division.

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:PATTERNBITS 6` sets the current mask number of serial pattern bits to 6.

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:PRESAMPBITS` 4 sets the current mask number of bits before the trigger pulse to 4.

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?
```

Arguments

`<NR1>` is an integer number that sets the record length.

Examples

`MASK:MASKPRE:RECORDLENGTH` 500 sets the current mask record length to 500 points.

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:TRIGTOSAMP 162E-9` sets the current mask time from the (leading edge) trigger position to the pulse bit sampling position to 162 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:VOFFSET 0.0` sets the current mask nominal vertical offset to 0.0 volts.

MASK:MASKPRE:VPOS

This command sets or returns the nominal vertical position, 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 in divisions.

Examples `MASK:MASKPRE:VPOS -0.8` sets the current mask nominal vertical position to -0.8 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:VSCALE 200E-3` sets the current mask nominal vertical scale to 200 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 bit width = 1 / (data rate of the signal).

Examples

`MASK:MASKPRE:WIDTH 648E-9` sets the current mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

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:POLARITY BOTH` sets pass/fail to test both positive (normal) and negative (inverted) waveforms.

MASK:SEG<m> (No Query Form)

This command deletes the specified mask segment from the Current mask.

Group

Mask

Syntax

```
MASK:SEG<m> DELEte
```

Arguments

<m> is an integer that specifies the mask segment number to delete from the Current mask.

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 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.

Group

Mask

Syntax

```
MASK:SEG<m>:NR_Pt?
```

Arguments

<m> is an integer number that specifies a mask segment number of the Current mask.

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. In other words, it sets or returns the vertices for a particular segment in the Current mask.

Group

Mask

Syntax

```
MASK:SEG<m>:POINTS <NR3>,<NR3>[,<NR3>,<NR3>]
MASK:SEG<m>:POINTS?
```

Arguments

`<m>` is an integer that specifies the Current mask segment number.

`<NR3>` refers to the coordinates of one of the vertices in the Current mask. Each pair of numbers is 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, then the value returned from the instrument will not be the same. If just one pair is input then they are 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:SOUrce

This command sets or reports which source will be compared against the mask(s) when counting is turned on. In other words, it controls which trace to use in mask counting. It also affects mask autoset 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:SOURCE CH1` 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` indicated 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_RCVDCLK2|SFI5_RCVDCLK3|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 (ANS1 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 (ANS1 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 (ANS1 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 (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)

FC1063E_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC1063E_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC1063E_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC1063E_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC1063E_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC1063E_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC1063E_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC1063E_AGR (ANS1 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 (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)

FC2125E_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)

FC2125E_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)

FC2125E_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)

FC2125E_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)

FC2125E_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)

FC2125E_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)

FC2125E_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)

FC2125E_AGR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)

FC4250E_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Beta, Transm)

FC4250E_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Delta, Transm)

FC4250E_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Gamma, Transm)

FC4250E_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Beta, Transm)

FC4250E_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Delta, Transm)

FC4250E_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Norm, Gamma, Transm)

FC4250E_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Beta, Recv)

FC4250E_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 4.0, Abs, Delta, Recv)

FC4250E_AGR (ANS1 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_RCVDCLK3 (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)

TFI5_3 (TFI-5, 3.1104 Gb/s)

USERMask

VIDEO270

VIDEO292M

VIDEO360

VSROC192 (VSR OC192/STM64, 1.24416 Gb/s)

Examples

MASK:STANDARD E1COAX 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. In other words, it controls whether the pass/fail test stops running when a violation is detected.

Group Mask

Syntax

```
MASK:STOPOnviolation {OFF|ON|<NR1>}
MASK:STOPOnviolation?
```

Arguments `ON` stops waveform acquisition on the first occurrence of a mask violation.

`OFF` turns off mask test stop on violation.

`<NR1>` = 0 turns off stop on violation; any other value turns on stop on violation.

Examples `MASK:STOPONVIOLATION ON` sets the instrument to stop waveform acquisition on the first occurrence of a waveform violation.

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 send a TTL signal out the Aux:Out port when the mask pass/fail status changes to 'Failing'. In other words, this command controls whether or not a TTL signal is sent out the AuxOut port on completion of the pass/fail test.

Group Mask

Syntax

```
MASK:TESt:AUX:COMPLetion {OFF|ON|<NR1>}
MASK:TESt:AUX:COMPLetion?
```

Arguments `ON` turns on the pass/fail mask:test:aux on completion.

`OFF` turns off 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:COMPLETION ON` turns on the pass/fail test on completion.

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 send a TTL signal out the AuxOut port when the pass/fail status changes to 'Failing'. In other words, it sends the signal when a failure occurs during the mask pass/fail test.

Group

Mask

Syntax

```
MASK:TESt:AUX:FAILure {OFF|ON|<NR1>}
MASK:TESt:AUX:FAILure?
```

Arguments

ON turns on the pass/fail mask:test:aux on failure.

OFF turns off the pass/fail mask:test:aux on failure. This is the default.

<NR1> = 0 disables the mask:test:aux:failure function; any other value enables it.

Examples

MASK:TEST:AUX:FAILURE ON turns on the pass/fail test on failure.

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

ON turns on the pass/fail beep on completion.

OFF turns off 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:COMPLETION ON turns on pas/fail beep on completion.

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 ‘Failed’. In other words, it sounds the bell when a failure occurs during the mask pass/fail test.

Group

Mask

Syntax

```
MASK:TESt:BEEP:FAILure {OFF|ON|<NR1>}
MASK:TESt:BEEP:FAILure?
```

Arguments

`ON` turns on the pass/fail beep on failure.

`OFF` turns off the pass/fail beep on failure. This is the default.

`<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:FAILURE OFF` turns off the pass/fail beep on failure.

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 prior to 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:DELAY 2.5` delays waveform evaluation by 2.5 seconds after the start of pass/fail testing.

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, using the current instrument hard copy settings, as soon as the pass/fail status changes to ‘Failing’. In other words, it prints the screen to the default printer.

Group

Mask

Syntax

```
MASK:TESt:HARDCopy {OFF|ON|<NR1>}
MASK:TESt:HARDCopy?
```

Arguments

`ON` turns on generate hard copy on failure.

`OFF` turns off generate hard copy on failure.

`<NR1>` = 0 turns off generate hard copy on failure, and any other value turns on generate hard copy on failure.

Examples

`MASK:TEST:HARDCOPY ON` turns on generate hard copy on failure.

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.

Group

Mask

Syntax

```
MASK:TESt:LOG:FAILure {OFF|ON|<NR1>}
MASK:TESt:LOG:FAILure?
```

Related Commands

MASK:TESt:SAVEWFM:FILEName

Arguments

`ON` turns on the pass/fail log on failure.

`OFF` turns off the pass/fail log on failure. This is the default.

`<NR1>` = 0 turns off the pass/fail log on failure, and any other integer turns on the pass/fail log on failure.

Examples

`MASK:TEST:LOG:FAILURE OFF` turns off the pass/fail log on failure.

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

ON turns on repeat on completion.

OFF turns off repeat on completion.

<NR1> = 0 turns off repeat on completion, and any other integer turns on repeat on completion.

Examples

MASK:TEST:REPEAT ON causes the instrument to continuously repeat the pass/fail test.

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:SAMPLE 100000 sets the instrument to acquire 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:THRESHOLD 100` sets the test to fail when 100 or more hits happen.

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.*

Group Mask

Syntax

```
MASK:TESt:SAVEWFM {OFF|ON|<NR1>}
MASK:TESt:SAVEWFM?
```

Arguments `ON` turns on copying all active channel waveforms to files.

`OFF` turns off 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.

Examples `MASK:TEST:SAVEWFM ON` sets the instrument to save the failing waveform data to files.

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).

Group

Mask

Syntax

```
MASK:TESt:SAVEWFM:FILEName <QString>
MASK:TESt:SAVEWFM:FILEName?
```

Related Commands

MASK:TESt:SAVEWFM, MASK:TESt:LOG:FAILure

Arguments

`<QString>` is a string representing the name of the file to save waveform data to

Examples

`MASK:TEST:SAVEWFM:FILENAME "TEST"` sets the file "TEST" as the file to save waveform data to on failure.

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

`ON` turns on the pass/fail SRQ on completion.

`OFF` turns off 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:COMPLETION ON` turns on pass/fail SRQ on completion.

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

ON turns on the pass/fail SRQ on failure.

OFF turns off the pass/fail SRQ on failure. This is the default.

<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:FAILURE OFF turns off the pass/fail SRQ on failure.

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

ON turns on mask pass/fail testing.

OFF turns off mask pass/fail testing.

<NR1> = 0 turns off mask pass/fail testing; any other integer turns on mask pass/fail testing.

Examples

MASK:TEST:STATE ON turns on mask pass/fail testing.

MASK:TESt:STATUS? (Query Only)

This 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`.

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'. In other words, it stops when a failure occurs during the mask pass/fail test. Repeat-on-completion mode has no effect.

Group

Mask

Syntax

```
MASK:TESt:STOP:FAILure {OFF|ON|<NR1>}
MASK:TESt:STOP:FAILure?
```

Arguments

`ON` turns on the pass/fail stop on failure.

`OFF` turns off the pass/fail stop on failure. This is the default.

`<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:FAILURE OFF` turns off the pass/fail stop on failure.

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:THRESHOLD ?` might return `:MASK:TEST:THRESHOLD 32`.

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. In other words, 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:WAVEFORM ?` might return

`:MASK:TEST:WAVEFORM 500`.

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:AMPLITUDE 2.4` sets the User mask nominal pulse amplitude to 2.4 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:BITRATE 156E6` sets the bit rate.

`MASK:USER:BITRATE ?` might return `:MASK:USER:BITRATE 1560000`.

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 40E-9 sets the nominal timing resolution used to draw the mask pulse shape 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:HTRIGPOS 0.6 sets the User mask nominal pulse leading edge position to the sixth horizontal graticule division.

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.

Group

Mask

Syntax

```
MASK:USER:LABel <string>
MASK:USER:LABel?
```

Arguments

`<string>` is an alphanumeric character string used to annotate a user mask label.

Examples

`MASK:USER:LABEL`l `"mikesdata"` assigns "mikesdata" as the user mask label.

`MASK:USER:LABEL` ? might return `:MASK:USER:LABel "mikesdata"` as the user mask label.

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.

Group

Mask

Syntax

```
MASK:USER:PATTERNBITS <NR1>
MASK:USER:PATTERNBITS?
```

Related Commands

MASK:USER:PRESAMPBITS

Arguments

`<NR1>` is an integer that sets the number of bits.

Examples

`MASK:USER:PATTERNBITS` 6 sets the User mask number of serial pattern bits to 6.

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:PRESAMPBITS 4` sets the User mask number of bits before the trigger pulse to 4.

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:RECORDLENGTH 500` sets the User mask record length to 500 points.

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.

Group Mask

Syntax

```
MASK:USER:SEG<m> DELEte
```

Arguments `<m>` is an integer that specifies the User mask segment number to delete from the User mask.

`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 command returns the number of points that make up the specified User mask segment. Each mask point consists of an X-Y pair of coordinates.

Group

Mask

Syntax

```
MASK:USER:SEG<m>:NR_Pt?
```

Arguments

<m> is an integer number that specifies a User mask segment number.

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. In other words, it sets or returns the vertices for a particular segment in the selected mask.

Group

Mask

Syntax

```
MASK:USER:SEG<m>:POINTS <NR3>, <NR3> [, <NR3>, <NR3>]
MASK:USER:SEG<m>:POINTS?
```

Arguments

<m> is an integer that specifies the User mask segment number.

<NR3> refers to the coordinates of one of the vertices in the User mask. Each pair of numbers is 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, then the value returned from the instrument will not be the same. If just one pair is input then they are 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: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: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:VPOS

This command sets or returns the nominal vertical position, 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 in divisions.

Examples

`MASK:USER:VPOS -0.8` sets the User mask nominal vertical position to -0.8 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:VSCALE 200E-3 sets the User mask nominal vertical scale to 200 millivolts 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:WIDTH 648E-9 sets the User mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

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.flt”` reads filter coefficients from the file and loads the filter for ARBFLT1.

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.

Group Math

Syntax `MATHArbflt<x>:READFile FORCe`

Arguments FORCE forces a reread of the filter file.

Examples `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.

Group Math

Syntax `MATHVAR?`

Related Commands MATHVAR:VAR<x>, MATH<x>:DEFine

Returns <NR3> are the stored numerical values.

Examples `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 reference 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".

Group

Math

Syntax

```
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.

Group

Math

Syntax

```
MATH<x>:DEFine <QString>
MATH<x>:DEFine?
```

Related Commands

MATHVAR:VAR<x>

Arguments

`<QString>` quoted string argument is the mathematical expression that defines the waveform.

Examples

`MATH2:DEFINE "CH1+CH2"` adds the Ch 1 waveform and Ch 2 waveform, storing the results in Math 2.

`MATH1 :DEFINE ?` might return

`:MATH1:DEFINE "CH2*REF2"` as the expression that defines Math 1.

MATH<x>:FILTer:MODe

This command or query sets or returns the filter risetime parameter.

Group

Math

Syntax

```
MATH<x>:FILTer:MODe {CENTered|SHIFted}
MATH<x>:FILTer:MODe?
```

Arguments

`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 on screen. Shifted mode is sometimes called a casual 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.

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 / (filter rise time). For a square wave input, the measurement system rise time of Math(x) = filter (chx) is very close to the filter rise time of Math(x).

Examples

`MATH1:FILTER:RISETIME 1.7e-3` sets the filter rise time to 1.7 milliseconds.

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 <string>
MATH<x>:LABel:NAMe?
```

Arguments

`<string>` specifies the label to annotate the math waveform.

Examples

`MATH2:LABEL:NAME "PROBE POINT7"` assigns "Probe point7" Math 2.

`MATH2:LABEL:NAME?` might return `:MATH2:LABEL:NAME "Probe point7"` indicating that Probe point 7 is the label for Math 2.

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 pixels) 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 500.

Examples

`MATH2:LABEL:XPOS 50` moves the waveform label for the MATH2 waveform (on a four-channel instrument) so that it begins 50 pixels to the right of the left edge of the screen.

`MATH2:LABEL:XPOS?` might return `:MATH2:LABEL:XPOS 50` indicating that the waveform label for the MATH2 waveform is currently 50 pixels 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 top edge of the screen. 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.

Group

Math

Syntax

```
MATH<x>:LABel:YPOS <NR1>
MATH<x>:LABel:YPOS?
```

Related Commands

MATH<x>:LABel:XPOS

Arguments

`<NR1>` is the location (in pixels) where the label for the selected math waveform is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 400.

Examples

`MATH2:LABEL:YPOS -25` moves the waveform label for the MATH2 waveform to just beneath (25 pixels below) the top of the screen.

`MATH2:LABEL:YPOS?` might return `:MATH2:LABEL:YPOS 0` indicating that the waveform label for the MATH2 waveform is currently located just beneath the top of the screen.

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 affect 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.

Group

Math

Syntax

```
MATH<x>:NUMAVg <NR1>
MATH<x>:NUMAVg?
```

Related Commands

ACQuire:NUMAVg

Arguments

`<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:MAIn: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 2-17: 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}
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 Math1, which causes the parameters for Math1 and Math2 to be locked together.

`MATH1:SPECTRAL:LOCK?` might return `:MATH1:SPECTRAL:LOCK 0` indicating that the parameter lock for Math1 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 Ω .

Examples

`MATH2:SPECTRAL:MAG DB` sets the SpectralMag units for Math2 to decibels.

`MATH2:SPECTRAL:MAG?` might return `:MATH2:SPECTRAL:MAG DB` indicating that the SpectralMag units for Math2 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 Math2 to degrees.

`MATH2:SPECTRAL:PHASE?` might return `:MATH2:SPECTRAL:PHASE RADIANS` indicating that the SpectralPhase units for Math2 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 × Log(y/<NR3>) where A is 10 if the input units are watts and A is 20 if the input units are otherwise.

`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>|FULl}
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:SUPPress

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

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}
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 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|TEKEXPone
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.

`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.

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

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>: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 Math2 operation will be called deg.

`MATH1:UNITSTRING?` might return `:MATH1:UNITSTRING "deg"` as the unit for the Math1 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 (min + 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 in divisions from the center graticule.

Examples

`MATH2:VERTICAL:POSITION 1.3E+00` positions the Math 2 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 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 (min + 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 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 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: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:DIRECTION FORWARDS` 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.*

Group

Measurement

Syntax

```
MEASUrement:IMMed:DELay:EDGE[1] {FALL|RISe}
MEASUrement:IMMed:DELay:EDGE[1]?
```

Related Commands

MEASUrement:IMMed:SOUrce<x>

Arguments

`FALL` specifies the falling edge.

`RISe` specifies the rising edge.

Examples

`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.

Group

Measurement

Syntax

```
MEASUrement:IMMed:DELay:EDGE2 {FALL|RISe}
MEASUrement:IMMed:DELay:EDGE2?
```

Related Commands

MEASUrement:IMMed:SOUrce<x>

Arguments

FALL specifies the falling edge.

RISe specifies the rising edge.

Examples

MEASUREMENT:IMMED:DELAY:EDGE2 RISE 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 This sets the high and low waveform levels statistically using a histogram algorithm.

MINMax This 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:NOISE HIGH 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:HIGH 1.71` 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:LOW 0.0` 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.*

Group

Measurement

Syntax

```
MEASUrement:IMMed:REFLevel:ABSolute:MID<x> <NR3>
MEASUrement:IMMed:REFLevel:ABSolute:MID<x>?
```

Related Commands

MEASUrement:IMMed:REFLevel:METHod

Arguments

`<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.

Examples

`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.

Group

Measurement

Syntax

```
MEASUrement:IMMed:REFLevel:METHod {ABSolute|PERCent}
MEASUrement:IMMed:REFLevel:METHod?
```

Arguments

`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.

Amplitude = High - 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.

Extinction dB = 10 × (log 10 (High / 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.

Extinction % = 100.0 × (Low / High)

EXTINCTRATIO 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 μW.

Extinction Ratio = (High / 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.

Negative Duty Cycle = (Negative Width) / Period × 100%

`NOVershoot` (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

Negative Overshoot = (Low - Minimum) / Amplitude × 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.

Crossing percent = 100 ×[(eye-crossing-point - PBASe)/(PTOP - 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.

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

Positive Overshoot = (Maximum - High) / Amplitude ×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 six × 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:TYPE FREQUENCY 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 HORizontal:MAIn:SCALe or CH<x>:SCALe will not necessarily have taken affect if 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.

Group

Measurement

Syntax

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

Examples

`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.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:DELay:DIREction {BACKWards|FORWards}
MEASUrement:MEAS<x>:DELay:DIREction?
```

Related Commands

MEASUrement:MEAS<x>:SOUrce<x>

Arguments

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 MEASUrement:MEAS<x>:DELay:EDGE<x> command to specify the slope of the edge.

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 MEASUrement:MEAS<x>:DELay:EDGE<x> command to specify the slope of the edge.

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 backwards.

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.

Group

Measurement

Syntax

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

Examples

`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*

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH <NR3>
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH?
```

Related Commands

MEASUrement:MEAS<x>:REFLevel:METHod

Arguments

`<NR3>` is the high reference level, in volts. The default is 0.0 V.

Examples

`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 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: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 only apply 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>}
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.

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.

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:SOUrce<x>:SIGType {PULse|EYE}
MEASUrement:MEAS<x>: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: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.

Group

Measurement

Syntax

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

Related Commands

MEASUrement:MEAS<x>:SOUrce<x>, MEASUrement:MEAS<x>:TYPe

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.

Amplitude = High - 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.

Extinction dB = 10 × (log 10 (High / 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.

Extinction % = 100.0 × (Low / High)

EXTINCTRATIO 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 μW.

Extinction Ratio = (High / 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.

Negative Duty Cycle = (Negative Width) / Period × 100%

`NOVershoot` (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

Negative Overshoot = (Low - Minimum) / Amplitude × 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.

Crossing percent = 100 ×[(eye-crossing-point - PBASe)/(PTOP - 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.

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

Positive Overshoot = (Maximum - High) / Amplitude ×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 six × 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:METHOD HISTOGRAM` 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:NOISE HIGH` 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:HIGH 1.71` 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.*

Group

Measurement

Syntax

```
MEASUrement:REFLevel:METHod {ABSolute|PERCent}
MEASUrement:REFLevel:METHod?
```

Arguments

`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).

`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.

Examples

`MEASUREMENT:REFLEVEL:METHOD ABSOLUTE` specifies that explicit user-defined values are used for the reference levels.

`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.

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:HIGH 95` 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:LOW 15` 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 only valid 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 all measurements off. 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 (μ) 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:MODE OFF turns statistic measurements off.

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.

Group

Measurement

Syntax

```
MEASUrement:STATIstics:WEIghting <NR1>
MEASUrement:STATIstics:WEIghting?
```

Related Commands

MEASUrement:STATIstics:MODe

Arguments

<NR1> is the time constant for the mean and standard deviation statistical accumulations.

Examples

MEASUREMENT:STATISTICS:WEIGHTING 4 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.

Group Miscellaneous

Syntax

```
NEWpass <QString>
```

Related Commands PASSWord, *PUD

Arguments `<QString>` is the new password, which can contain up to 10 characters.

Examples `NEWPASS "mypassword"` creates a new password (mypassword) for accessing your protected data.

*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The *OPC? query places the ASCII character "1" into the output queue when all such OPC commands are complete. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The *OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods.

Table 2-18: Commands the 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)
Hardcopy operation	HARDCopy START
Calibration step	Refer to the Service Manual.

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.

Group

Miscellaneous

Syntax

```
PASSWord <QString>
```

Related Commands

NEWpass, *PUD

Arguments

`<QString>` is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.

Examples

`PASSWORD "XYZZY"` enables the *PUD and NEWPass set commands.

`PASSWORD` disables the *PUD and NEWPass set commands. You can still use the query version of *PUD.

*PSC

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

Group

Status and Error

Syntax

```
*PSC {OFF|ON|<NR1>}
*PSC?
```

Related Commands

DESE, *ESE, FACtory, *RST, *SRE

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

`*PSC 0` sets the power-on status clear flag to false.

`*PSC ?` might return 1 to indicate that the power-on status clear flag is set to true.

*PUD

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

Group

Status and Error

Syntax

```
*PUD {<Block>|<QString>}
*PUD?
```

Related Commands

PASSWord

Arguments

`<Block>` is a block containing up to 100 characters.

`<QString>` is a string containing up to 100 characters.

Examples

`*PUD #229This instrument belongs to me` stores the string "This instrument belongs to me" in the user protected data area.

`*PUD ?` might return `#221PROPERTY OF COMPANY X`.

*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:SETUP FACTORY` recalls (and makes current) the instrument setup to its factory defaults.

`RECALL:SETUP 2` 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: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.

Group

Save and Recall

Syntax

```
RECAll:WAVEform <file path>,REF<x>
```

Related Commands

DELEte:WAVEform, SAVe:WAVEform

Arguments

`REF<x>` specifies a location in internal reference memory. Reference memory location values range from 1 through 4.

`<file path>` specifies a location for a stored waveform 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 waveform from the default directory (C:\TekScope|Waveforms). <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.

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

Arguments None.

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 to "My Reference".

`REF3:LABEL:NAME?` might return `:REF3:LABEL:NAME "Signal2"` indicating that the label name for Reference 3 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 (in pixels) 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 500.

Examples

`REF4:LABEL:XPOS 10` moves the waveform label for the REF3 waveform, so that it begins 10 pixels to the right of the left edge of the screen.

`REF2:LABEL:XPOS?` might return `:REF2:LABEL:XPOS 150` indicating that the x-axis for the REF 2 waveform is currently 150 pixels 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 top 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 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 pixels) where the waveform label for the selected reference is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 385.

Examples

`REF3:LABEL:YPOS -10` moves the waveform label for the REF 3 waveform to just beneath (10 pixels) the top of the screen.

`REF2:LABEL:YPOS?` might return `:REF2:LABEL:YPOS 0` indicating that the waveform label for the REF 2 waveform is currently located just beneath the top of the screen.

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

Arguments

`<NR3>` is the desired position, 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 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 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 as well as 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 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 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: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?
```

Related Commands ROSc:STATE?

Arguments `INTERnal` specifies the internal 10 MHz crystal oscillator as the time base reference.

`EXTernal` specifies the user-supplied external signal as the time base reference.

Examples `ROSC:SOURCE INTERNAL` specifies the internal 10 MHz crystal oscillator as the time base reference.

`ROSC:SOURCE ?` might return `:ROSC:SOURCE INTERNAL` indicating that the 10 MHz crystal oscillator is being used as the time base reference.

ROSc:STATE? (Query Only)

This query-only command returns whether the time base reference oscillator is locked. This command will return either LOCKED or UNLOCKED.

Group Miscellaneous

Syntax

```
ROSc:STATE?
```

Related Commands

ROSc:SOUrce

Examples

`ROSC:STATE?` might return `:ROSC:STATE LOCKED` indicating that the time base reference is locked.

*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings. The *RST command does not alter 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.

Group Save and Recall

Syntax

```
*SAV <NR1>
```

Related Commands *RCL, RECAll:SETUp, SAVe:SETUp

Arguments `<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.

Examples `*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.

Group Mask

Syntax

```
SAVe:MASK <QString>
```

Related Commands RECAll:MASK

Arguments `<QString>` is the location to where the mask is stored. <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 is stored in ASCII format and is user-editable.

Examples `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.

Group Save and Recall

Syntax

```
SAVe:SETUp {<file path>|<NR1>}
```

Related Commands *RCL, RECAll:SETUp, *SAV

Arguments `<file path>` is the target location for storing the 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 store the file in 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 are highly recommended. For setups, use the extension ".SET".

`<NR1>` 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.

Examples `SAVE:SETUP 5` saves the current instrument setup in memory location 5.

`SAVE:SETUP "TEK00000.SET"` saves the current instrument setup in the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).

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.

Group Save and Recall

Syntax

```
SAVe:WAVEform <wfm>,{<file path>|REF<x>}
```

Related Commands

DELEte:WAVEform, RECAll:WAVEform

Arguments

`<wfm>` is the waveform that will be saved. Valid waveforms include CH<x>, MATH<y>, and `REF<x>`.

`REF<x>` specifies the location in internal reference memory to where the waveform is stored.

`<file path>` is the location to where the waveform is stored. <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 waveform from 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. The instrument will generate an error if you use any other extension for saving a waveform.

Examples

`SAVE:WAVEFORM MATH2,REF2` saves the Math2 waveform in reference memory location2.

`SAVE:WAVEFORM MATH1,"TEK0000.WFM"` saves the Math1 waveform to the file TEK00000.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
{INTERNal|MATHCad|MATLab|SPREADSHEETCsv|SPREADSHEETTxt}
SAVe:WAVEform:FILEFormat?
```

Related Commands

CURVe, DATa, DATa:STARt, DATa:STOP, SAVe:WAVEform, WFMInpre:NR_Pt, WFMOutpre: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.

Examples

`SAVE:WAVEFORM:FILEFORMAT INTERNAL` specifies that the internal file format is the format used for saving waveforms.

`SAVE:WAVEFORM:FILEFORMAT ?` might return `:SAVE:WAVEFORM:FILEFORMAT INTERNAL` 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 `*SDS 2` changes setup slot 2 to reference the factory setup.

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: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>?
```

Arguments

ON displays the indicated channel waveform.

OFF turns off the display of the indicated channel waveform.

<NR1> = 0 turns off the display of the specified channel waveform; any other value turns on the display of the specified waveform.

Examples

SELECT:CH1 ON displays channel 1.

SELECT:CH1? might return SELECT:CH1 1 indicating channel 1 is displayed.

SELect:CONTROl <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:CONTROL CH2 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:<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>?
```

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 REF1 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;:CMDBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;:DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST 500.0000E-3;: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:NAME 1, "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.

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.

Group

Miscellaneous

Syntax

```
TEKSecure
```

Examples

`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. *Turning off both DIAg:CONTROL:HALT and DIAg:CONTROL:LOOP before executing the TEST command is recommended.*

Group

Diagnostics

Syntax

```
TEST <QString>
TEST?
```

Related Commands

TEST:RESults?, TEST:RESults:VERBose?, TEST:STOP

Arguments

`<QString>` sets the test ID, which ranges from 0 through 3 characters. If no test ID is specified, all available diagnostics are executed.

Examples

`TEST "1"` executes all Acquisition tests (Subsystem:Acquisition).

`TEST "11"` executes all Acquisition Memory tests (Subsystem:Acquisition, Area:Memory)

`TEST "113"` executes a specific Acquisition Memory test (Subsystem:Acquisition, Area:Memory, Test:Address Lines).

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.

Group

Diagnostics

Syntax

```
TEST:RESults?
```

Related Commands

TEST, TEST:RESults:VERBose?

Examples

`TEST:RESULTS?` might return `:TEST:RESULTS "135"` indicating the specific failure code of the first test failure.

`TEST:RESULTS?` might return `:TEST:RESULTS "PASS"` indicating that the previously executed test passed.

`TEST:RESULTS?` might return `:TEST:RESULTS "*****"` indicating that the test was not run.

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 `TRIGGER FORCE` 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:INPUT:CH1
HIGH;CH2 HIGH;CH3 HIGH;CH4
HIGH;: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: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

Arguments

`AC` selects AC trigger coupling, which passes the input signals above 60 Hz to the trigger circuitry.

`DC` selects DC trigger coupling, which passes all input signals to the trigger circuitry.

`HFRej` coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

`LFRej` coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.

`NOISErej` coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may require greater trigger signal amplitude.

`ATRIGger` this B trigger command sets the B trigger coupling to match the setting on the A trigger.

Examples

`TRIGGER:A:EDGE:COUPLING DC` sets the A edge trigger coupling to DC.

`TRIGGER:A:EDGE:COUPLING?` might return `:TRIGGER:A:EDGE:COUPLING DC` indicating that the A edge trigger coupling is set to DC.

TRIGger:{A|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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:COUPling:CH<x>
{AC|DC|HFRej|LFRej|NOISErej}
TRIGger:{A|B}:EDGE:COUPling:CH<x>?
```

Related Commands

TRIGger:{A|B}:EDGE:COUPling

Arguments

`AC` selects AC trigger coupling.

`DC` selects `DC` trigger coupling.

`HFRej` selects high frequency low sensitivity.

`LFRej` selects low frequency low sensitivity.

`NOISErej` selects `DC` low sensitivity.

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|EIther}
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.

EITHER specifies to trigger on either the rising or falling 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 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: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}
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 connector located on the rear panel of the instrument.

`CH<x>` specifies one input channel as the edge trigger source.

`LINE` specifies AC line voltage, and is for A Trigger only.

Examples

`TRIGGER:A:EDGE:SOURCE CH1` sets channel 1 as the A edge trigger source.

`TRIGGER:A:EDGE:SOURCE?` might return `:TRIGGER:A:EDGE:SOURCE CH1` indicating that channel 1 is the A edge trigger source.

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:LVLSRCPREFERENCE 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:CLAss

This command sets or queries the class of the Logic Trigger. Used in conjunction with the TRIGger:A: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: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`.

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`

which indicates 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 three are "don't care."

TRIGger:{A|B}:LOGIc:INPut:CH<x>

This command sets or queries the A logical input for the logic trigger 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: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` argument sets the instrument to trigger when the pattern becomes true.

`FALSe` argument sets the instrument to trigger when the pattern becomes false.

`LESSThan` argument 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:SETHold?
```

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 1.4000;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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:SOUrce CH<x>
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:SOUrce?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:CLOCk:EDGE, TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold

Arguments

`CH<x>` specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two channel instruments.

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold {ECL|TTL|<NR3>}
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:CLOCk:EDGE, TRIGger:{A|B}:LOGIc:SETHold:CLOCk:SOUrce

Arguments

`ECL` specifies a preset `ECL` high level.

`TTL` specifies a preset `TTL` high level.

`<NR3>` is the clock level, in volts.

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold:CH<x>
{ECL|TTL|<NR3>}
TRIGger:{A|B}:LOGIc:SETHold:CLOCk:THReshold:CH<x>?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:CLOCk:EDGE, TRIGger:{A|B}:LOGIc:SETHold:CLOCk:SOUrce

Arguments

`ECL` specifies a preset ECL high level.

`TTL` specifies a preset TTL high level.

`<NR3>` is the clock level, in volts.

Examples

`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:SETHold:DATa?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:CLOCk?

Examples

`TRIGGER:A:LOGIC:SETHOLD:DATA?` might return `:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.4000;SOURCE CH1` 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:SETHold:DATa:SOUrce CH<x>
TRIGger:{A|B}:LOGIc:SETHold: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>

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: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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:HOLDTime <NR3>
TRIGger:{A|B}:LOGIc:SETHold:HOLDTime?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:SETTime

Arguments

`<NR3>` specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Examples

`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 or Logic in the Trigger if Setup/Hold drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:LOGIc:SETHold:QUAlify?
```

Arguments

`OCCurs` specifies a trigger if any detectable event occurs.

`LOGIc` This argument specifies a trigger if the individual channel qualifications meet the logic patterns and thresholds set by the TRIGger:{A|B}:LOGIc:INPut:CH<x> commands.

Examples

`TRIGGER:A:LOGIC:SETHOLD:QUALIFY OCCURS` sets the trigger qualifier to OCCurs, which initiates a trigger if the instrument detects any setup and hold trigger qualifications.

`TRIGGER:A:LOGIC:SETHOLD:QUALIFY?` might return :TRIGGER:A:LOGIC:SETHOLD:QUALIFY OCCURS indicating that a trigger will occur if the instrument detects any of the setup and hold qualifications.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:SETHold:SETTime <NR3>
TRIGger:{A|B}:LOGIc:SETHold:SETTime?
```

Related Commands

TRIGger:{A|B}:LOGIc:SETHold:HOLDTime

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 µs.

`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`.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:STATE:WHEn {TRUe|FALSe}
TRIGger:{A|B}:LOGIc:STATE:WHEn?
```

Related Commands

TRIGger:{A|B}:LOGIc:INPut:CH<x>, TRIGger:{A|B}:LOGIc:STATE:INPut:CH<x>

Arguments

`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.

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:THReshold?
```

Related Commands

TRIGger:{A|B}:LOGIc:THReshold:CH<x>, TRIGger:{A|B}:LOGIc:INPut?, TRIGger:{A|B}:LOGIc:INPut:CH<x>

Examples

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:LVLSrcpreferenceTRIGger: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: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: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, , , andTRIGger:{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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:GLItch?
```

Related Commands

TRIGger:{A|B}:PULse:GLItch:POLarity, TRIGger:{A|B}:PULse:GLItch:TRIGIF, TRIGger:{A|B}:PULse:GLItch:WIDth

Examples

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:GLItch:POLarity
{POSITIVe|NEGAtive|EITher}
TRIGger:{A|B}:PULse:GLItch:POLarity?
```

Related Commands

TRIGger:{A|B}:LEVel, TRIGger:{A|B}:PULse:GLItch:WIDth

Arguments

`POSITIVe` specifies that the instrument will only trigger when the polarity of the glitch is positive.

`NEGAtive` specifies that the instrument will only trigger when the polarity of the glitch is negative.

`EITher` specifies that the instrument will trigger when the polarity of the glitch is either positive or negative.

Examples

`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.

`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.

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

`POSITIVE` specifies that the instrument will only trigger when the polarity of the glitch is positive.

`NEGATIVE` specifies that the instrument will only trigger when the polarity of the glitch is negative.

`EITHER` specifies that the instrument will trigger when the polarity of the glitch is either positive or negative.

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 or Logic in the Trigger if Glitch drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:GLItch:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:GLItch:QUAlify?
```

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.

Examples

`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.

Group Trigger

Syntax

```
TRIGger:{A|B}:PULse:GLItch:TRIGIF {ACCept|REJect}
TRIGger:{A|B}:PULse:GLItch:TRIGIF?
```

Related Commands TRIGger:{A|B}:PULse:GLItch:WIDth

Arguments `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 µs.

`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:LOGIC:INPUT:CH1 HIGH; CH2 X; CH3 X; CH4 LOW;:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH11.2000; 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|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.

Examples

`TRIGGER:A:PULSE:RUNT:POLARITY NEGATIVE` specifies that the polarity of the A pulse runt trigger is 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 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

`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:CH3 NEGATIVE` This command specifies that the polarity of the channel 3 A pulse runt trigger is negative.

`TRIGGER:A:PULSE:RUNT:POLARITY:CH3?` This query might return :TRIGGER:A:PULSE:RUNT:POLARITY:CH3 POSITIVE indicating that the polarity of the A pulse runt trigger is 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 or Logic in the Trigger if Runt drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:RUNT:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:RUNT:QUAlify?
```

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:

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:RUNT:THReshold?
```

Related Commands

TRIGger:{A|B}:PULse:RUNT:POLarity, TRIGger:{A|B}:PULse:RUNT:THReshold:BOTh, TRIGger:{A|B}:PULse:RUNT:THReshold:HIGH, TRIGger:{A|B}:PULse:RUNT:THReshold:LOW

Examples

`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 800 mV.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:RUNT:THReshold:BOTh {TTL|ECL}
```

Related Commands

TRIGger:{A|B}:PULse:RUNT:THReshold?

Arguments

`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.

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:RUNT:THReshold:HIGH <NR3>
TRIGger:{A|B}:PULse:RUNT:THReshold:HIGH?
```

Related Commands TRIGger:{A|B}:PULse:RUNT:THReshold?

Arguments `<NR3>` specifies the threshold value, in volts.

Examples `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:THReshold:LOW <NR3>
TRIGger:{A|B}:PULse:RUNT:THReshold: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 µs.

`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>
TRIGger:{A|B}:PULse:SOUrce?
```

Related Commands

TRIGger:{A|B}:EDGE:SOUrce

Arguments

`CH<x>` specifies one of the input channels, which range from 1 through 4.

Examples

`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 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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TIMEOut?
```

Related Commands

TRIGger:{A|B}:PULse:TIMEOut:POLarity, TRIGger:{A|B}:PULse:TIMEOut:TIMe

Examples

`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.

Group

Trigger

Syntax

```
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 or Logic in the Trigger if Timeout drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TIMEOut:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:TIMEOut:QUAlify?
```

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:C<x> and TRIGGER:{A|B}:LOGIC:THRESHOLD:CH<x> commands.

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 µs.

`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: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 µs.

`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

Arguments

`POSITIVe` indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

`NEGative` indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

`EITher` indicates either positive or negative polarity.

Examples

`TRIGGER:A:PULSE:TRANSITION:POLARITY NEGATIVE` sets the transition polarity to negative.

`TRIGGER:A:PULSE:TRANSITION:POLARITY?` might return `:TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER` indicating that the polarity can be either positive or negative.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TRANsition:POLarity:CH<x>
{EITher|NEGAtive|POSITIVe}
TRIGger:{A|B}:PULse:TRANsition:POLarity:CH<x>?
```

Related Commands

TRIGger:{A|B}:PULse:TRANsition:DELTATime

Arguments

`POSITIVE` indicates that a pulse edge must traverse from the lower (most negative) to higher (post positive) level for transition triggering to occur.

`NEGATIVE` indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

`EITHER` indicates either positive or negative polarity.

Examples

`TRIGGER:A:PULSE:TRANSITION:DELTATIME NEGATIVE` sets the trigger A transition polarity to negative.

`TRIGGER:A:PULSE:TRANSITION:POLARITY?` might return :TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER indicating that the transition can be either positive or negative for trigger A.

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 or Logic in the Trigger if Violation drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TRANsition:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:TRANsition:QUAlify?
```

Arguments

`OCCURS` specifies a trigger if any violations occur.

`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.

Examples

`TRIGGER:A:PULSE:TRANSITION:QUALIFY OCCURS` sets the trigger qualifier to OCCurs, which initiates an A trigger if the instrument detects any transition violations.

`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.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TRANsition:THReshold?
```

Related Commands

TRIGger:{A|B}:PULse:TRANsition:DELTATime, TRIGger:{A|B}:PULse:TRANsition:POLarity

Examples

TRIGGER:A:PULSE:TRANSITION:THRESHOLD? might return `:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 1.2000;LOW 800.0000E-3` indicating the upper and lower threshold limits for the transition time trigger.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:TRANsition:THReshold:BOTh {TTL|ECL}
```

Related Commands

TRIGger:{A|B}:PULse:TRANsition:THReshold:HIGH, TRIGger:{A|B}:PULse:TRANsition:THReshold:LOW

Arguments

`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.

Examples

`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:DELTATime, 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 µs.

`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 µs.

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 a positive pulse.

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

Syntax

```
TRIGger:{A|B}:PULse:WIDth:POLarity:CH<x> {NEGAtive|POSITIVe}
TRIGger:{A|B}:PULse:WIDth:POLarity:CH<x>?
```

Related Commands

TRIGger:{A|B}:PULse:WIDth?

Arguments

NEGATIVE specifies a negative pulse.

POSITIVE specifies a positive pulse.

Examples

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 a positive pulse 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 or Logic in the Trigger if Width drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:WIDth:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:WIDth:QUAlify?
```

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 TIRGGER:{A|B}:LOGIC:THRESHOLD:CH<x> commands.

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.

Group Trigger

Syntax

```
TRIGger:{A|B}:PULse:WINdow?
```

Examples `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`.

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.

Group Trigger

Syntax

```
TRIGger:{A|B}:PULse:WINdow:EVENT
{OUTSIDEGreater|INSIDEGreater|ENTERSWindow|EXITSWindow}
TRIGger:{A|B}:PULse: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

`TRIGGER:A:PULSE:WINDOW:WHEN OCCURS` sets the widow 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:QUAlify

This command sets or queries the Window Trigger qualification. This is equivalent to selecting Window Setup from the Trig menu and selecting Occurs or Logic in the Trigger if Window drop-down list box.

NOTE. *If you use the LOGIc argument, you must specify the channel logic conditions.*

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:WINdow:QUAlify {OCCurs|LOGIc}
TRIGger:{A|B}:PULse:WINdow:QUAlify?
```

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.

Examples

`TRIGGER:A:PULSE:WINDOW:QUALIFY OCCURS` This command 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?` This query 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?
```

Arguments

`<NR3>` specifies the threshold value in volts.

Examples

`TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW 0.8` sets the lower threshold of the window trigger to 0.8 V.

`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.

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.

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULse:WINdow:WIDTH <NR3>
TRIGger:{A|B}:PULse:WINdow:WIDTH?
```

Arguments

`<NR3>` argument specifies the minimum width in seconds.

Examples

`TRIGGER:A:PULSE:WINDOW:WIDTH 15E-9` sets the minimum width of a valid window violation to 15 ns.

`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.

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: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:A SETLEVEL` 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:INPUT:CH1
HIGH;CH2 HIGH;CH3 HIGH;CH4
HIGH;: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: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.

`FRAME` 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 identifer 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 indication an end of file condition.

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.

WRITEsets the CAN data direction to WRITE.

NOCAREsets the CAN data direction to either.

Examples

TRIGGER:A:CAN:DATA:DIRECTION WRITE sets the CAN data direction to Write.

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 ±5.0.

Examples

TRIGGER:A:CAN:DATA:LEVEL 0.0 sets the CAN trigger data level to 0.0 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 {CH1|CH2|CH3|CH4}
TRIGger:A:CAN:DATa:SOUrce?
```

Arguments

CH<x> specifies the source for the CAN trigger.

Examples

TRIGGER:A:CAN:DATA:SOURCE CH1 sets the CAN trigger data source to channel one.

TRIGger:A:CAN:DATa:VALue

CAN option only: This command sets or quires the binary data string used for CAN Trigger if the trigger condition is ID or IDANDDATA.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:CAN:DATa:VALue <String>
TRIGger:A:CAN:DATa:VALue?
```

Arguments `String` is up to 32 bits specifying the CAN data value.

Examples `TRIGGER:A:CAN:DATA:VALUE "1011"` sets the CAN data value to 1011.

TRIGger:A:CAN:FORMat

CAN option only: This command sets or queries the CAN data format.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:CAN:FORMat {BINary|HEX}
TRIGger:A:CAN:FORMat?
```

Arguments `BINARY` specifies binary as the CAN data format.

`HEX` specifies hexadecimal as the CAN data format.

Examples `TRIGGER:A:CAN:FORMAT BINARY` sets the CAN trigger data format to binary.

TRIGger:A:CAN:FRAMEtype

CAN option only: This command sets or queries the CAN trigger frame type.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:CAN:FRAMEtype {DATA|REMote}
TRIGger:A:CAN:FRAMEtype?
```

Arguments

DATA specifies a data frame type.

REMOTE specifies a remote frame type.

ERROR specifies an error frame type.

OVERLOAD specifies an overload frame type.

Examples

TRIGGER:A:CAN:FRAMETYPE DATA sets the CAN trigger frame type to DATA.

TRIGger:A:CAN:IDENTifier:MODe

CAN option only: This command sets or queries the CAN trigger identifier mode.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:CAN:IDENTifier:MODe {STANdard|EXTENded}
TRIGger:A:CAN:IDENTifier:MODe?
```

Arguments

STANDARD sets the CAN trigger identifier mode to standard mode.

EXTENDED sets the CAN trigger identifier mode to extended mode.

Examples

TRIGGER:A:CAN:IDENTIFIER:MODE STANDARD sets the CAN trigger identifier mode to standard.

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.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:CAN:IDENTifier:VALue <String>
TRIGger:A:CAN:IDENTifier:VALue?
```

Arguments

String 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: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 trigger two trigger on CAN systems with a bit rate of 33K.

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

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:
HIGH <NR3>
TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:
HIGH?
```

Arguments

`<NR3>` is the high threshold parameter expressed in volts.

Examples `TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH 2.8E-2` sets the threshold high level to 2.8E-2.

`TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH?` might return `:TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH 2.8E-2` indicating the threshold high level in volts.

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.

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}:THReshold:
LOW <NR3>
TRIGger:A:COMMunication:{AMI|HDB3|B3ZS|B6ZS|B8ZS}:THReshold:
LOW?
```

Arguments `<NR3>`

The high threshold parameter expressed in volts.

Examples `TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW 2.8E-2` sets the threshold low level to 2.8E-2.

`TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW?` might return `:TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW 2.8E-2` indicating the threshold low level in volts.

TRIGger:A:COMMunication:BITRate

This command sets or queries the bit rate.

Conditions This command is only valid if the instrument has Option MTM is installed

Group Trigger

Syntax

```
TRIGger:A:COMMunication:BITRate <NR3>
TRIGger:A:COMMunication:BITRate?
```

Arguments

<NR3> is a nonnegative number greater than one and expressed as bits per second.

Examples

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

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:CLOCk:POLarity {RISe|FALL}
TRIGger:A:COMMunication: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:COMMUNICATION:CLOCK:POLARITY RISE 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.

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:CMI:PULSEForm
{PLUSOne|MINUSOne|ZERO|EYEdiagram}
TRIGger:A:COMMunication:CMI:PULSEForm?
```

Arguments

`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.

Examples

`TRIGGER:A:COMMUNICATION:CMI:PULSEFORM PLUSONE` 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.

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:CODe
{AMI|HDB3|B3ZS|B6ZS|B8ZS|CMI|NRZ|MLT3|MANChester}
TRIGger:A:COMMunication:CODe?
```

Arguments

`AMI`

`HDB3`

`B3ZS`

`B6ZS`

B8ZS

CMI

NRZ

MLT3

MANChester

Examples TRIGGER:A:COMMUNICATION:CODE AMI selects 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 only valid 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:SOURCE CH1 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.

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:SOUrce:TYPe {DATa|CLOCk|RECOVered}
TRIGger:A:COMMunication:SOUrce:TYPe?
```

Arguments

`DATA` causes the instrument to trigger and shift five unit intervals to form the expected eye pattern.

`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.

Examples

`TRIGGER:A:COMMUNICATION:SOURCE:TYPE CLOCK` sets the source type to clock.

`TRIGGER:A:COMMUNICATION:SOURCE:TYPE ?` might return `:TRIGGER:A:COMMUNICATION:SOURCE:TYPE DATA` indicating that the source type is data.

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 do post processing on AMI isolated pulses, and pulse width settings if CMI.

Conditions

This command is only valid if the instrument has Option MTM is installed.

Group

Trigger

Syntax

```
TRIGger:A:COMMunication:STANdard
{ATAG1|ATAG2|ATAG3|CLOCKCoax|
CLOCKSymmetrical|Custom|D1|D2|DS0Contra|
DS0Double| DS0Single|DS0Timing|DS1|DS1A|
DS1C|DS2| DS2RATECoax|DS2RATESymmetrical|
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 2-19: 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

Table 2-19: Communication Trigger Standards, (cont.)

Argument	Code	Bit Rate	Mask Type	Models
DS2RATE Symmetrical	AMI B6ZS	6.312 Mb/s	ITU-T G.703	All
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	

Table 2-19: Communication Trigger Standards, (cont.)

Argument	Code	Bit Rate	Mask Type	Models
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	
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	≥500MHz
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 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` showing that the holdoff time is set to 4 µs.

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.

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:TIME 10` sets the A trigger holdoff time to 10 s.

`TRIGGER:A:HOLDOFF:TIME ?` might return `:TRIGGER:A:HOLDOFFTIME 1.2000E-06` indicating that the A trigger holdoff time is set to 1.2 µs.

TRIGger:A: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: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: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 <String>
TRIGger:A:I2C:ADDRess:VALue?
```

Arguments `String` is up to 7 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: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 ±5.

Examples `TRIGGER:A:I2C:CLOCK:LEVEL 0.0` sets the I2c clock level to 0.0.

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` specifies the I2C clock source is channel 1.

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` specifies start as the I2C trigger condition.

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 specifies write as the I2C data direction.

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

Syntax

```
TRIGger:A:I2C:DATa:LEVel <NR3>
TRIGger:A:I2C:DATa:LEVel?
```

Arguments NR3 specifies the I2C data level with a range of ±%.0.

Examples TRIGGER:A:I2C:DATA:LEVEL 0.0 sets the I2C data level to 0.0.

TRIGger:A:I2C:DATa:SOUrce

DPO7000 Series only: This command sets or queries the data source for the I2C serial trigger.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:I2C:DATa:SOUrce CH<x>
TRIGger:A:I2C:DATa:SOUrce?
```

Arguments CH<x> specifies the data source. X can be 1 through 4.

Examples TRIGGER:A:I2C:DATA:SOURCE CH1 specifies channel 1 as the data source for the I2C serial trigger.

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.

Group

Low Speed Serial Trigger

Syntax

```
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 of 0.

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 <String>
TRIGger:A:I2C:DATa:VALue?
```

Arguments

`String` 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: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:LOGIc? (Query Only)

This query-only command returns all of the logic trigger parameters.

Group Trigger

Syntax

```
TRIGger:A: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: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|NORMal}
TRIGger:A:MODe?
```

Related Commands TRIGger:{A|B}:LEVel

Arguments

AUTO generates a trigger if one is not detected within a specified time period.

NORMal waits for a valid trigger event.

Examples

TRIGGER:A:MODE NORMAL specifies that a valid trigger event must occur before a trigger is generated.

TRIGGER:A:MODE ? might return :TRIGGER:A:MODE NORMAL indicating that a valid trigger event must occur before a trigger is generated.

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: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 ±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: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 2.

TRIGger:A:RS232:DATa:VALue

DPO7000 Series only: This command sets or queries the binary data string used for RS232 triggering.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:RS232:DATa:VALue <String>
TRIGger:A:RS232:DATa:VALue?
```

Arguments

NR3 specifies the RS232 data value. The value can be up to 8 bits.

Examples

`TRIGGER:A:RS232:DATA:VALUE “1011”` sets the RS232 triggering data value to 1011.

TRIGger:A:RS232:FORMat

DPO7000 Series only: This command sets or queries the display format for the RS232 data value.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:RS232:FORMat {BINary|HEX}
TRIGger:A:RS232:FORMat?
```

Arguments

BINARY specifies binary for the display format.

HEX specifies hexadecimal for the display format.

Examples

TRIGGER:A:RS232:FORMAT HEX sets the display format for the RS232 data to hexadecimal.

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:PARITY EVEN sets the RS232 parity to even.

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 51.8400E+6 indicating a bit rate of 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 32.0000E-3 indicating that the clock level is 32 mV.

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` This command sets RISE as the clock polarity.

`TRIGger:A:SERIAL:CLOCK:POLARITY?` This query might return RISE indicating that the polarity is RISE.

TRIGger:A:SERIAL:CLOCk:SOUrce

This command sets or queries the serial data source channel.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:SERIAL:CLOCk:SOUrce {CH1|CH2|CH3|CH4|RECOVered}
TRIGger:A:SERIAL:CLOCk:SOUrce?
```

Arguments

`CH1-4` specifies one of the input channels.

`RECOVERED` specifies clock polarity and level are extracted from the serial data stream and other clock parameters are ignored.

Examples

`TRIGger:A:SERIAL:CLOCK:SOURCE CH1` This command sets channel 1 as the source channel.

`TRIGger:A:SERIAL:CLOCK:SOURCE?` This query might return CH1 indicating that the source is channel 1.

TRIGger:A:SERIAL:CODe

This command sets or queries the signal code.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:SERIAL:CODe {NRZ|S8B10B}
TRIGger:A:SERIAL:CODe?
```

Arguments

`NRZ` sets the code to NRZ.

`S8B10B` sets the code to S8B0B.

Examples

`TRIGger:A:SERIAL:CODE NRZ` sets the signal code to NRZ.

`TRIGger:A:SERIAL:CODe?` might return NRZ indicating that the signal code is NRZ.

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 BINARY indicating that the Pattern string is in a binary 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 <string>
TRIGger:A:SERIAL:DATa:PATtern?
```

Arguments

`STRING` 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: `XXXXXXXA` 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: `XXXXXXXXXXXXXXXXXXXXXXXXXXXX1010` showing the response due to the binary format. Four binary characters ”`1010`” now replace the hex character ”A”.

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

`CH1-4` 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 CH1 indicating that the source is channel 1.

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|ENET1
```

```
TRIGger:A:SERIAL:STANdard?
```

Related Commands

Arguments

All of the standards may not be available on every instrument.

`CUSTom`: NRZ, 155.5 Mb/s

`ENET100FX`: MLT3, 100 Mb/s

`ENET1250`: NRZ, 1.25 Gb/s

`FC133`: NRZ, 132.8 Mb/s

`FC266`: NRZ, 265.6 Mb/s

`FC531`: NRZ, 531.2 Mb/s

`FC1063`: NRZ, 1.063 Gb/s

`FW1394BS400B`: NRZ, 491.5 Mb/s

`FW1394BS800B`: NRZ, 983.0 Mb/s

`OC1`: NRZ, 51.84 Mb/s

`OC3`: NRZ, 155.5 Mb/s

`OC12`: NRZ, 622.1 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

`VSROC192`: NRZ, 1.2441 Gb/s, triggers VSR OC192/STM64

Examples

`TRIGger:A:SERIAL:STANDARD OC1` sets the standard to OC1.

`TRIGger:A:SERIAL:STANDARD?` might return OC1 Indicating that the signal standard is OC1.

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}
TRIGger:A:SPI:CONDition?
```

Arguments

MISO specifies the MISO condition for SPI triggering.

MOSI specifies the MISI condition for SPI triggering.

MISOMOSI specifies the either the MISO or MISI condition for SPI triggering.

Examples

`TRIGGER:A:SPI:CONDITION MOSI` sets the SPI trigger condition 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:LEVel

DPO7000 Series only: This command sets or queries the threshold for the SPI MISO data source.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MISO:LEVel <NR3>
TRIGger:A:SPI:DATa:MISO:LEVel?
```

Arguments

NR3 specifies the threshold for the SPI trigger MISO data source. The range is ±5.0.

Examples

`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:SOUrce

DPO7000 Series only: This command sets or queries the MISO data source for the SPI trigger.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MISO:SOUrce CH<x>
TRIGger:A:SPI:DATa:MISO:SOUrce?
```

Arguments

CH<x> specifies the source for the SPI trigger MISO data source. X can be 1, 2, 3, or 4.

Examples

`TRIGGER:A:SPI:DATA:MISO:SOURCE CH2` sets the SPI trigger MISO data source to channel 2.

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.

Group

Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MISO:VALue <String>
TRIGger:A:SPI:DATa:MISO:VALue?
```

Arguments

String 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: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:LEVel

DPO7000 Series only: This command sets or queries the threshold for the SPI MOSI data source.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MOSI:LEVel <NR3>
TRIGger:A:SPI:DATa:MOSI:LEVel?
```

Arguments NR3 specifies the threshold for the SPI trigger MOSI data source. The range is ±5.0.

Examples 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:MOSI:SOUrce

DPO7000 Series only: This command sets or queries the MOSI data source for the SPI trigger.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MOSI:SOUrce CH<x>
TRIGger:A:SPI:DATa:MOSI:SOUrce?
```

Arguments CH<x> specifies the source for the SPI trigger MOSI data source. X can be 1, 2, 3, or 4.

Examples `TRIGGER:A:SPI:DATA:MOSI:SOURCE CH2` sets the SPI trigger MOSI data source to channel 2.

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.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:DATa:MOSI:VALue <String>
TRIGger:A:SPI:DATa:MOSI:VALue?
```

Arguments String 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: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: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: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 FALLING sets the SPI SCLK polarity to falling.

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: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 2.

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:LEVel

DPO7000 Series only: This command sets or queries the threshold for the SPI trigger Slave Select (SS) signal.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:SS:LEVel <NR3>
TRIGger:A:SPI:SS:LEVel?
```

Arguments NR3 specifies the threshold for the SPI trigger SS line. The threshold range is ±5.0.

Examples `TRIGGER:A:SPI:SS:LEVEL 0.0` sets the threshold level for the SPI trigger SS to 0.0.

TRIGger:A:SPI:SS:SOUrce

DPO7000 Series only: This command sets or queries the SPI trigger Slave Select (SS) source.

Group Low Speed Serial Trigger

Syntax

```
TRIGger:A:SPI:SS:SOUrce CH<x>
TRIGger:A:SPI:SS:SOUrce?
```

Arguments CH<x> specifies the source for the SPI SS signal. X can be 1, 2, 3, or 4.

Examples `TRIGGER:A:SPI:SS:SOURCE CH2` sets channel 2 as the source for SS.

TRIGger:A:TYPe

This command sets or queries the type of 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.

Group Trigger

Syntax

```
TRIGger:A:TYPe {EDGE|LOGIc|PULse|VIDeo|
I2C|CAN|SPI|COMMunication|SERIAL|RS232}
TRIGger:A: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.

Examples

TRIGGER:A:TYPE EDGE 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:VIDeo? (Query Only)

DPO7000 Series only: This query-only command returns the A trigger video parameters.

Group Trigger

Syntax

```
TRIGger:A:VIDeo?
```

Examples `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.

Group Trigger

Syntax

```
TRIGger:A:VIDeo:CUSTom?
```

Examples `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.

Group Trigger

Syntax

```
TRIGger:A:VIDeo:CUSTom:FORMat {INTERLAced|PROGressive}
TRIGger:A:VIDeo:CUSTom:FORMat?
```

Arguments `INTERLAced` argument sets the format for interlaced video lines.

`PROGressive` argument sets the format for progressive video lines.

Examples `TRIGGER:A:VIDEO:CUSTOM:FORMAT PROGRESSIVE` 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.

Group

Trigger

Syntax

```
TRIGger:A:VIDeo:CUSTom:LINEPeriod <NR3>
TRIGger:A:VIDeo:CUSTom:LINEPeriod?
```

Arguments

<NR3> is the time for each horizontal scan line.

Examples

`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.

Group

Trigger

Syntax

```
TRIGger:A:VIDeo:CUSTom:SYNCInterval <NR3>
TRIGger:A:VIDeo:CUSTom:SYNCInterval?
```

Arguments

<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.

Group Trigger

Syntax

```
TRIGger:A:VIDeo:HOLdoff:FIELD <NR3>
TRIGger:A:VIDeo:HOLdoff:FIELD?
```

Arguments `<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.

Examples `TRIGGER:A:VIDEO:HOLDOFF:FIELD 4.5` 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.

Group Trigger

Syntax

```
TRIGger:A:VIDeo:LINE <NR1>
TRIGger:A:VIDeo:LINE?
```

Related Commands TRIGger:A:VIDeo:FIELD

Arguments `<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 2-20: Video Line Numbering Ranges

Video Standard	Line Number Range
CUSTOM	4–3000
NTSC	1–263 (odd) and 264–525 (even)
PAL	1–625

Table 2-20: Video Line Numbering Ranges, (cont.)

Video Standard	Line Number Range
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:LINE 23` 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|NORMal}
TRIGger:A:VIDeo:POLarity?
```

Arguments

`INVERTed` argument sets the instrument to trigger on a positive video sync pulse.

`NORMAl` argument sets the instrument to trigger on a negative video sync pulse.

Examples

`TRIGGER:A:VIDEO:POLARITY NORMAL` 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. This command is the same as the command.

Group

Trigger

Syntax

```
TRIGger:A:VIDeo:SCAN {RATE1|RATE2|RATE3|RATE4|RATE5}
TRIGger:A:VIDeo:SCAN?
```

Arguments

RATE1 argument sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.

RATE2 argument sets the range of the video line scan rate to 20 kHz through 25 kHz.

RATE3 argument sets the range of the video line scan rate to 25 kHz through 35 kHz.

RATE4 argument sets the range of the video line scan rate to 35 kHz through 50 kHz.

RATE5 argument sets the range of the video line scan rate to 50 kHz through 65 kHz.

Examples

TRIGGER:A:VIDEO:SCAN RATE1 sets the scan rate of the A video trigger to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).

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.

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.

Group

Trigger

Syntax

```
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:SOURCE CH1 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|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:SYNCIntervalcommand.

TRILevelcustom argument sets the instrument to use custom video horizontal scan rate parameters that you set with the TRIGger:A:VIDeo:CUSTom:LINEPeriodcommand.

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.

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:STANDARD NTSC` sets the instrument to trigger on NTSC-standard video signals.

`TRIGGER:A:VIDEO:STANDARD ?` might return `:TRIGger:A:VIDeo:STANDARD NTSC` 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:B SETLEVEL` 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.*

Group

Trigger

Syntax

```
TRIGger:B:BY {EVENTS|TIMe}
TRIGger:B:BY?
```

Related Commands

TRIGger:B:EVENTS:COUNt, TRIGger:B:TIMe, HORizontal[:MAIn]:DELay:MODe, HORizontal[:MAIn]:DELay:TIMe TRIGger:B:STATE

Arguments

`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 TRIGger:B:`EVENTS`:COUNt.

`TIMe` sets the B trigger to occur a set time after the A trigger event. The time period is specified by TRIGger:B:`TIMe`.

Examples

`TRIGGER:B:BY TIME` sets the B trigger to occur at a set time after the A trigger event.

`TRIGGER:B:BY` ? might return `:TRIGGER:B:BY EVENTS` indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

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.

Group

Trigger

Syntax

```
TRIGger:B:EVENTS?
```

Related Commands

TRIGger:B:EVENTS:COUNt, TRIGger:B:STATE

Examples

`TRIGGER:B:EVENTS?` might return

`:TRIGGER:B:EVENTS:COUNT 2`

indicating that 2 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:COUNT 4` specifies that the B trigger will occur four trigger events after the A trigger.

`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.

`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:THReshold {ECL|TTL|<NR3>}
TRIGger:B:RESET:THReshold?
```

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` This command 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: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` argument indicates that the B trigger is active and in causes trigger events conjunction with the A trigger.

`OFF` argument 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:STATE ON` 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:TIME 4E-6` sets the B trigger delay time to 4 µs.

`TRIGGER:B:TIME ?` might return `:TRIGGER:B:TIME 16.0000E-9` indicating that the B trigger time is set to 16 ns.

TRIGger:B:TYPe

This command sets or queries the type of B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing Edge.

Group Trigger

Syntax

```
TRIGger:B:TYPe {EDGE|PULse|LOGIc}
TRIGger:B:TYPe?
```

Related Commands TRIGger:A:TYPe TRIGger:B:STATE

Arguments `EDGE` sets the B trigger type to edge.

`PULSE` specifies that a trigger occurs when a specified pulse is found and is controlled by the `TRIGger:B:PULse` commands.

`LOGIC` specifies that a trigger occurs when specified conditions are met and is controlled by the `TRIGger:B:LOGIc` commands.

Examples `TRIGGER:B:TYPE EDGE` sets the B trigger type to edge.

`TRIGGER:B:TYPE` ? might return `:TRIGGER:B:TYPE EDGE`.

TRIGger:LVLSrcpreference

This command sets or queries the dependent source/level trigger feature. SRCIndependent is the default value.

Group Trigger

Syntax

```
TRIGger:LVLSrcpreference {SRCDependent|SRCIndependent}
TRIGger:LVLSrcpreference?
```

Related Commands TRIGger:{A|B}:LEVel:CH<x>,

Arguments `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.

Examples `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: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 true, which returns full-length keywords for applicable setting queries.

`ON` sets the Verbose state to false, which returns minimum-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

`VERBOSE ON` enables the Verbose state.

`VERBOSE ?` might return `:VERB 0` indicating that the Verbose state is disabled.

*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,Sampl mode";
:CURVE51,50,51,48,51,48,50,49,51,49,51,48,51,48,51,49,50,49,50,48,49,49
```

WAVFRMStream? (Query Only)

This query only command returns WFMQUTPRE? 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 returns the number of bits per binary waveform point for the waveform, as specified by the DATa:DESTination command. This specification is only meaningful 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, WFMOutpre:BIT_Nr

Arguments

`<NR1>` number of bits per data point can be 8, 16 (RI, RP) or 32 (FP).

Examples

`WFMINPRE:BIT_NR 16` 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

WFMOutpre: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_FMT FP` 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 returns 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, WFMOutpre: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_NR 1` 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_OR MSB` 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`II 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:ENCDG ASCii` 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_Pt

This command sets or returns the number of data points that are in the transmitted waveform record.

Group

Waveform Transfer

Syntax

```
WFMInpre:NR_Pt <NR1>
WFMInpre:NR_Pt?
```

Related Commands

CURVe, DATa, DATa:STARt, DATa:STOP, SAVe:WAVEform, SAVe:WAVEform:FILEFormat, , , WFMOutpre:NR_Pt?

Arguments

`<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.

Examples

`WFMINPRE:NR_PT 5000` 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:

Xn = XZEro + XINcr (N-PT_Off)

Ynmax = YZEro + YMUlt (ynmax - YOFf)

Ynmin = 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:

Xn = XZEro + XINcr (N-PT_Off)

Yn = YZEro + YMUlt (Yn - YOFf)

Examples

WFMINPRE:PT_FMT ENV 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 returns 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?
```

Related Commands

DATa:DESTination, DATa:STARt, WFMOutpre:PT_Off?

Arguments

<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.

Examples

WFMINPRE:PT_OFF 1 specifies that the trigger point is the same as the Data:Start point.

WFMINPRE:PT_OFF ? might return :WFMINPRE:PT_OFF 0 indicating that the incoming waveform trigger point is the first point in the waveform record.

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.

Group

Waveform Transfer

Syntax

```
WFMInpre:WFId <QString>
```

Related Commands

DATa:DESTination, WFMInpre:BN_Fmt, WFMInpre:ENCdg, WFMOutpre:WFId?

Arguments

<QString> must be a valid IEEE-488.2 string (but the contents are ignored).

Examples

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.

Group

Waveform Transfer

Syntax

```
WFMInpre:XINcr <NR3>
WFMInpre:XINcr?
```

Related Commands

WFMInpre:XUNit, WFMOutpre:XINcr?

Arguments

<NR3> is the horizontal interval representation.

Examples

`WFMINPRE:XINCR 3E-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 returns 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:XZERO 5.7E-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:YMUlt

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:YMUlt <NR3>
WFMInpre:YMUlt?
```

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:YMULT 20E-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:YOFF 50 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:YZERO 1.5E+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 µV below the center of the data range (given that WFMInpre:YUNit is set to V).

WFMOutpre? (Query Only)

This query-only command returns 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 returns 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_NR 16` sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

`WFMOUTPRE:BIT_NR ?` might return `:WFMOUTPRE:BIT_NR 8` indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

WFMOutpre:BN_Fmt

This command sets or queries the format of binary data for outgoing waveforms specified by the DATa:SOUrce command.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BN_Fmt {RI|RP|FP}
WFMOutpre:BN_Fmt?
```

Related Commands

DATa:SOUrce

Arguments

`RI` specifies signed integer data point representation.

`RP` specifies positive integer data point representation.

`FP` specifies single-precision binary floating point data point representation.

Examples

`WFMOUTPRE:BN_FMT FP` specifies that outgoing waveform data will be in single-precision binary floating point format.

`WFMOUTPRE:BN_FMT ?` might return `:WFMOUTPRE:BN_FMT RI` indicating that the outgoing waveform data is currently in signed integer format.

WFMOutpre:BYT_Nr

This command sets or returns 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_NR 1` sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

`WFMOUTPRE:BYT_NR ?` might return `:WFMOUTPRE:BYT_NR 2` indicating that there are 2 bytes per outgoing waveform data point.

WFMOutpre:BYT_Or

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BYT_Or {LSB|MSB}
WFMOutpre:BYT_Or?
```

Related Commands

WFMOutpre:ENCdg

Arguments

`LSB` specifies that the least significant byte will be transmitted first.

`MSB` specifies that the most significant byte will be transmitted first.

Examples

`WFMOUTPRE:BYT_OR MSB` sets the most significant outgoing byte of waveform data to be transmitted first.

`WFMOUTPRE:BYT_OR?` might return `:WFMOUTPRE:BYT_OR LSB` indicating that the least significant data byte will be transmitted first.

WFMOutpre:ENCdg

This command sets and queries the type of encoding for outgoing waveforms.

Group

Waveform Transfer

Syntax

```
WFMOutpre:ENCdg {ASCii|BINary}
WFMOutpre:ENCdg?
```

Related Commands

DATa:ENCdg, WFMOutpre:BYT_Nr, WFMOutpre:BYT_Or, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt

Arguments

`ASCii` specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.

`BINary` specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT_Nr, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt and WFMOutpre:BYT_Or.

Examples

`WFMOUTPRE:ENCDG ASCii` specifies that the outgoing waveform data will be sent in ASCII format.

`WFMOUTPRE:ENCDG?` might return `:WFMOUTPRE:ENCDG BINary` indicating that outgoing waveform data will be sent in binary format.

WFMOutpre:NR_Pt? (Query Only)

This query-only command returns the number of points for the DATa:SOUrce waveform that will be transmitted in response to a CURVe? query.

Group

Waveform Transfer

Syntax

```
WFMOutpre:NR_Pt?
```

Related Commands CURVe, DATa, DATa:STARt, DATa:STOP, SAVe:WAVEform, SAVe:WAVEform:FILEFormat, , WFMInpre:NR_Pt,

Examples `WFMOUTPRE:NR_PT?` might return `:WFMOUTPRE:NR_PT 5000` indicating that there are 5000 data points to be sent.

WFMOutpre:PT_Fmt? (Query Only)

This query-only command returns the point format for the waveform specified by the DATa:SOUrce command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the CURVe data points.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax

```
WFMOutpre:PT_Fmt?
```

Related Commands CURVe, DATa:SOUrce

Examples `WFMOUTPRE:PT_FMT?` might return `:WFMOutpre:PT_Fmt ENV` indicating that the waveform data is a series of min-max pairs.

WFMOutpre:PT_Off? (Query Only)

This query-only command returns the trigger point relative to DATa:STARt for the waveform specified by the DATa:SOUrce command.

NOTE. *This returned value is the point immediately following the actual trigger.*

Group Waveform Transfer

Syntax

```
WFMOutpre:PT_Off?
```

Related Commands DATa:SOUrce, DATa:STARt, WFMOutpre:XZEro?

Examples

`WFMOUTPRE:PT_OFF?` might return `:WFMOUTPRE:PT_OFF 251` specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:PT_ORder? (Query Only)

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 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` specifying 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 tables below:

Table 2-21: 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"
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 µs/point (500 µs/div).

WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group

Waveform Transfer

Syntax

```
WFMOutpre:XUNit?
```

Related Commands

DATa:SOUrce

Examples

`WFMOUTPRE:XUNIT?` might return `:WFMOUTPRE:XUNIT "HZ"` indicating that the horizontal units for the waveform are in Hertz.

WFMOutpre:XZEro? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATa:SOUrce waveform does not exist.

***NOTE.** During (i.e., ACQuire:STATE OFF), this is the only preamble that changes on each acquisition. If a query is run during steady state operation (that is, all control changes have settled and triggers are arriving on a regular basis), the XZEro value of the last stopped state is returned.*

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 `ZOOM RESET` resets the zoom transforms to default values for all traces.

`ZOOM` ? might return

```
:ZOOM:MODE 0;GRATICULE:SIZE 80;SPLIT
EIGHTYTWENTY: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:POSTION0.0000;
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:SIZE` 100 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: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:MODE OFF` turns off Zoom mode.

`ZOOM:MODE` ? might return `:ZOOM:MODE 1` indicating that Zoom mode is currently turned on.

ZOOm:SCROLL:DIREction

This command sets or returns 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:DIRECTION FWD 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 returns the state of Scroll Lock, which "locks" zoomed waveforms under common control.

Group

Zoom

Syntax

```
ZOOm:SCROLL:LOCk {ON|OFF|NR1}
ZOOm:SCROLL:LOCk?
```

Arguments

ON enables Scroll Lock.

OFF disables Scroll Lock.

<NR1> = 0 disables Scroll Lock for all zoom windows; any other value enables Scroll Lock for all zoom windows.

Examples

ZOOM:SCROLL:LOCK ON enables Scroll Lock for zoomed waveforms identified as locked.

ZOOM:SCROLL:LOCK ? might return :ZOOM:SCROLL:LOCK OFF indicating that the scroll lock function is disabled.

ZOOm:SCROLL:SPEED

This command sets or returns the speed of automatic scrolling.

Group Zoom

Syntax

```
ZOOm:SCROLL:SPEED <NR1>
ZOOm:SCROLL:SPEED?
```

Arguments <NR1> is a value from 1 to 10.

Examples `ZOOM:SCROLL:SPEED 2` will scroll zoomed waveform(s) 2 divisions per update.

`ZOOM:SCROLL:SPEED ?` might return `:ZOOM:SCROLL:SPEED 1` indicating that the scroll speed is set to move zoomed waveform(s) 1 division per update.

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? 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>: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.

Group

Zoom

Syntax

```
ZOOm:ZOOM<x>:SCROLLLock {ON|OFF|1|0|NR1}
ZOOm:ZOOM<x>:SCROLLLock?
```

Arguments

`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>.

Examples

`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.

Group

Zoom

Syntax

```
ZOOm:ZOOM<x>:STATE {ON|OFF|<NR1>}
ZOOm:ZOOM<x>:STATE?
```

Arguments

`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.

Examples

`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>: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.

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.

Group

Zoom

Syntax

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAle <NR3>
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAle?
```

Arguments

`<NR3>` is the amount of expansion in the horizontal direction in 1-2-5 increments.

Examples

`ZOOM:ZOOM1:CH1:HORIZONTAL:SCALE 5` sets the horizontal scale factor of Channel 1 to 5.

`ZOOM:ZOOM2:CH2:HORIZONTAL:SCALE?` might return `:ZOOM2:CH2:HORIZONTAL:SCALE 1`, indicating that Channel 2 has a horizontal scale factor of 1.

ZOOm:ZOOM<x>:<wfm>:HORizontal:TIMEPosition

This command sets or queries the zoom horizontal position of the specified waveform for the specified zoom, but in time from the trigger instead of percent of waveform.

Group

Zoom

Syntax

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:TIMEPosition <NRF>
ZOOm:ZOOM<x>:<wfm>:HORizontal:TIMEPosition?
```

Arguments

`NRF` specifies the horizontal position in time from the trigger.

Returns

`ZOOm:ZOOM<x>:<wfm>:HORizontal:TIMEPosition` 1e-9 sets the zoom reference pointer at 1 ns in the acquired waveform.

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.

Group

Zoom

Syntax

```
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition <NR3>
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition?
```

Arguments

`<NR3>` is the vertical position, expressed in divisions.

Examples

`ZOOM:ZOOM1:CH1:VERTICAL:POSITION 2` sets the Zoom1 vertical position to 2, which centers the zoom trace at the second division of the acquired trace.

`ZOOM:ZOOM1:CH1:VERTICAL:POSITION?` might return `:ZOOm1:CH1:VERTical:POSition 0.0000`, 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.

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 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.

7	6	5	4	3	2	1	0
—	RQS 6 MSS	ESB	MAV	—	—	—	—

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 below.

The Device Event Status Enable Register (DESER). This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

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

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."

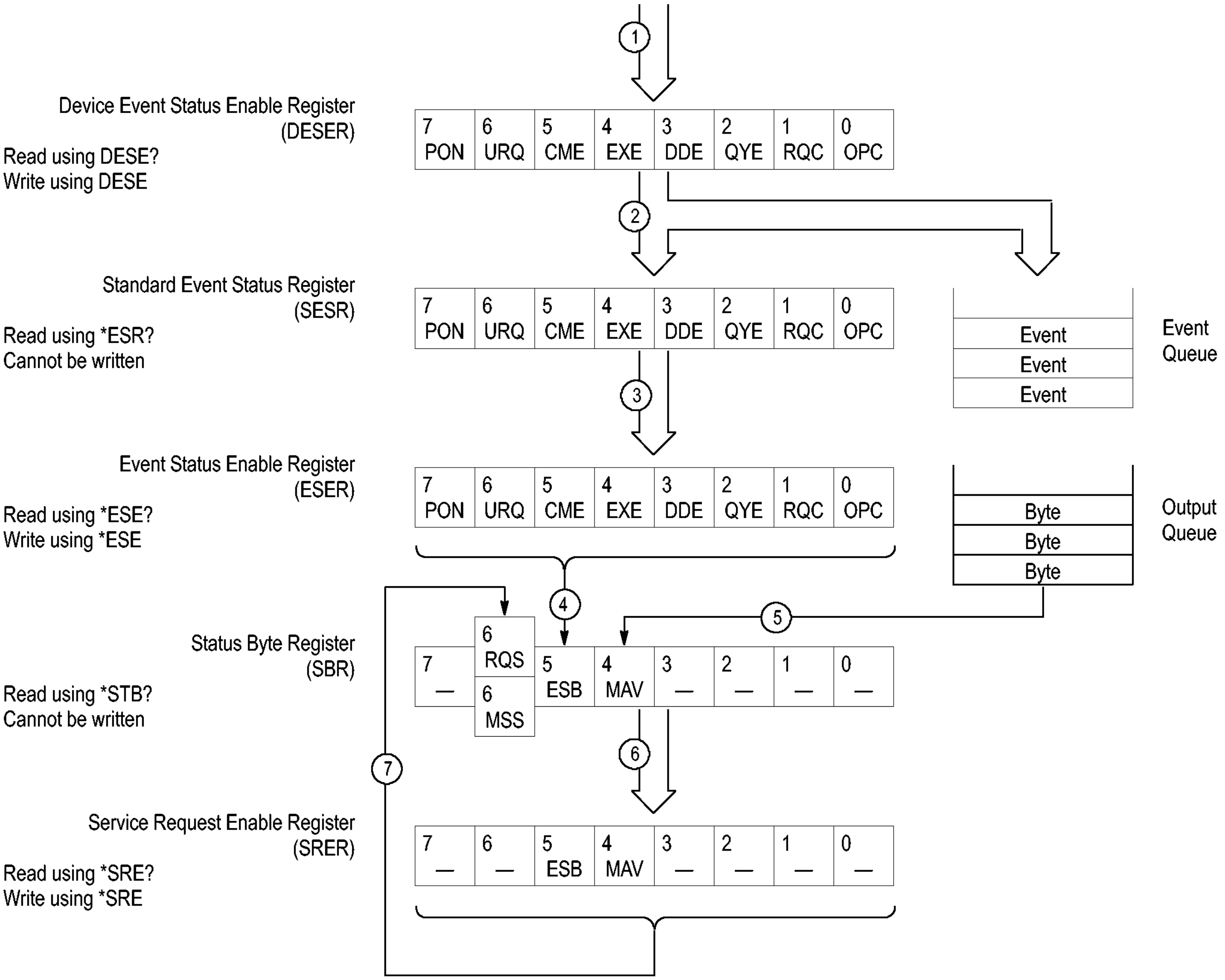
Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

Event Handling Sequence

The 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

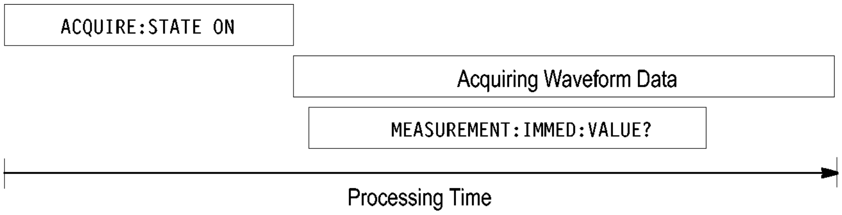
Although most GPIB commands are completed almost immediately after being received by the instrument, some commands start a process that requires more time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may be a few seconds 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.

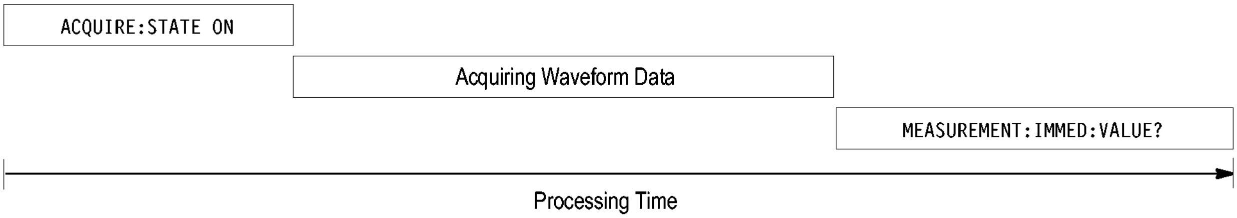
For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```
/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL: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: MEAS1: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 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: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: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: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: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: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

*/

*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)

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
109	Missing parameter
110	Command header error
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 number
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

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)

Code	Message
200	Execution error
201	Invalid while in local
202	Settings lost due to RTL
210	Trigger error
211	Trigger ignored
212	Arm ignored
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
230	Data corrupt or stale
240	Hardware error
241	Hardware missing
244	Invalid parameter selected
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
260	Execution expression error
261	Math error in expression
2200	Measurement error, Measurement system error
2201	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

Table 3-5: Execution Error Messages (EXE Bit 4), (cont.)

Code	Message
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref Crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing
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
2241	Waveform request is invalid
2243	This measurement cannot be performed on this type of waveform
2244	Source waveform is not active
2248	This ref cannot be activated
2249	Reference deletion error, Waveform in use for math
2250	Reference error, Waveform reference 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
2259	File too big
2261	Calibration error, wait for the warmup interval to expire before invoking SPC
2400	Not enough memory available
2401	This channel cannot be activated
2402	Math/Meas/Histo have circular definition
2410	Empty math string

Table 3-5: Execution Error Messages (EXE Bit 4), (cont.)

Code	Message
2411	Syntax error in math string
2412	Semantic error in math string
2413	Math expression is too complex
2420	Histogram cannot be performed on this type of waveform

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
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)

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
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

Internal Warning. The following table shows internal errors that indicate an internal fault in the instrument.

Table 3-9: Internal Warning Messages

Code	Message
600	Internal warning
630	Internal warning, 50 Ω overload

Programming Examples

Overview

The example programs illustrate methods you can use to control the instrument. These programs are included on the Product Software CD. These example programs are installed as part of the *GPIB Programmer* installation, which includes the Programmer Online Guide.

To install the "GPIB Programmer", perform the procedure outlined in the product software CD manual.

An Examples directory will be created with the following path name:*C:\Program Files\Tektronix\TekScope\documentation\bin\Programmer\Examples*.

Within the *Examples* directory are two subdirectories, *Source* and *Programs*.

Source contains the source files, written in Microsoft Visual C++ , Version 6.0, which is required to build executable files for each of the following example programs:

- Meas.c – This example program demonstrates how to perform a periodic amplitude measurement on CH 1 of your instrument.
- GetWfm.c – This example program demonstrates how to acquire and output the CH 1 waveform in a 500-point RIBANARY format. The program then queries the instrument to get the waveform preamble information, formats the binary waveform data as ASCII values, and, finally, writes out a report of the waveform preamble and ASCII data points to a file named WFM_DATA.PRN.
- Tl.c – This example program demonstrates how to use a general talker/listener program to allow the user to send commands and queries to the instrument and display the responses.

A *README* file in the *Source* directory has the latest documentation.

Programs contains compiled, executable files for each of the example programs in the Source directory (see above).

The programs run on a PC-compatible system equipped with a National Instruments GPIB board and associated drivers.

Compiling GPIB Programs

The example GPIB programs make the following assumptions:

- The GPIB controller board is "GPIB0" (board 0). If you have more than one GPIB controller in your workstation, and you want to use a GPIB controller other than board 0, you must edit the source files accordingly.
- The instrument is connected to the GPIB controller (see above) and is set to address 1 (DEV1). If you want to use another address, then you must edit the source files accordingly.

Each program requires adding the following source files to the associated VC++ project (which you will create).

- *GPIBERR.C*
- *GPIBREAD.C*
- *GPIBWAIT.C*
- *GPIBWRIT.C*

In addition to the above source files, you will also need to add the following files provided by National Instruments:

- *decl-32.h*
- *gpib-32.obj*

Visual C++

To make an executable for any of the example programs in the Source directory, perform the following steps:

1. Install Microsoft Visual C^{++}, Version 6.0.

2. Install the National Instruments GPIB board and drivers.

3. Copy the following source files from your National Instruments GPIB drivers directory to this directory ("Examples\Source"):

- *decl-32.h*
- *pib-32.obj*

4. Create a new "Win32 Console Application" project in Visual C++ located in this directory ("Examples\Source"). For example, if you want to build the Talker/Listener application, select this directory for the location, and enter a suitable name for the project, such as TL.

5. Add the following Visual C++ source files to the project:

- *..\GPIBERR.C*
- *..\GPIBREAD*
- *..\GPIBWAIT.C*
- *..\GPIBWRIT.C*

6. Add the following source files provided by National Instruments to the project:

- *..\decl-32.h*
- *..gpib-32.obj*

7. Add the appropriate program main source file to the project (see Overview for a list of the three source files).

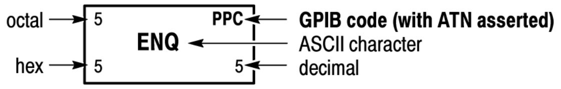
8. Build and test the project.

9. To build another of the example projects, repeat steps 4 through 8.

Appendix A: Character Set

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 LA0 SP 20 32	60 LA16 0 30 48	100 TA0 @ 40 64	120 TA16 P 50 80	140 SA0 ` 60 96	160 SA16 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 LA1 ! 21 33	61 LA17 1 31 49	101 TA1 A 41 65	121 TA17 Q 51 81	141 SA1 a 61 97	161 SA17 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 LA2 " 22 34	62 LA18 2 32 50	102 TA2 B 42 66	122 TA18 R 52 82	142 SA2 b 62 98	162 SA18 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 LA3 # 23 35	63 LA19 3 33 51	103 TA3 C 43 67	123 TA19 S 53 83	143 SA3 c 63 99	163 SA19 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 LA4 $ 24 36	64 LA20 4 34 52	104 TA4 D 44 68	124 TA20 T 54 84	144 SA4 d 64 100	164 SA20 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 LA5 % 25 37	65 LA21 5 35 53	105 TA5 E 45 69	125 TA21 U 55 85	145 SA5 e 65 101	165 SA21 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 LA6 & 26 38	66 LA22 6 36 54	106 TA6 F 46 70	126 TA22 V 56 86	146 SA6 f 66 102	166 SA22 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 LA7 ' 27 39	67 LA23 7 37 55	107 TA7 G 47 71	127 TA23 W 57 87	147 SA7 g 67 103	167 SA23 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 LA8 (28 40	70 LA24 8 38 56	110 TA8 H 48 72	130 TA24 X 58 88	150 SA8 h 68 104	170 SA24 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51 LA9) 29 41	71 LA25 9 39 57	111 TA9 I 49 73	131 TA25 Y 59 89	151 SA9 i 69 105	171 SA25 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 LA10 * 2A 42	72 LA26 : 3A 58	112 TA10 J 4A 74	132 TA26 Z 5A 90	152 SA10 j 6A 106	172 SA26 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 LA11 + 2B 43	73 LA27 ; 3B 59	113 TA11 K 4B 75	133 TA27 [5B 91	153 SA11 k 6B 107	173 SA27 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 LA12 , 2C 44	74 LA28 < 3C 60	114 TA12 L 4C 76	134 TA28 \ 5C 92	154 SA12 l 6C 108	174 SA28 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 LA13 – 2D 45	75 LA29 = 3D 61	115 TA13 M 4D 77	135 TA29] 5D 93	155 SA13 m 6D 109	175 SA29 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 LA14 . 2E 46	76 LA30 > 3E 62	116 TA14 N 4E 78	136 TA30 ^ 5E 94	156 SA14 n 6E 110	176 SA30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 LA15 / 2F 47	77 UNL ? 3F 63	117 TA15 O 4F 79	137 UNT – 5F 95	157 SA15 o 6F 111	177 RUBOUT (DEL) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY



Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling. For the most robust code, use the full spelling, since spelling rules may change over time and among instrument models.

```
*CAL
*CLS
*DDT
*ESE
*ESR
*IDN
*LRN
*OPC
*OPT
*PSC
*PUD
*RCL
*RST
*SAV
*SDS
*SRE
*STB
*TRG
*TST
*WAI
A
ABSolute
AC
ACCept
ACKMISS
ACQDURATION
ACQLENGTH
ACQuire
ACTIVE
ACTUal
ADDR10
ADDR7
ADDRANDDATA
ADDRess
ALIas
ALL
ALLEv
ALLFields
ALLLines
ALLMatched
ALLOcate
AMI
AMPlitude
AND
APPLYtoall
AREA
AREa
ASCIi
ASCii
ATAG1
ATAG2
ATAG3
ATARXG1
ATARXG2
ATARXG3
ATATXG1
ATATXG2
ATATXG3
ATRIGger
ATTEN1000X
ATTEN100X
ATTEN10X
ATTEN125X
ATTEN12_5X
ATTEN150X
ATTEN15X
ATTEN175X
ATTEN17_5X
ATTEN1X
ATTEN1_25X
ATTEN1_5X
ATTEN1_75X
ATTEN200X
ATTEN20X
ATTEN250X
ATTEN25X
ATTEN2X
ATTEN2_5X
ATTEN300X
ATTEN30X
ATTEN350X
ATTEN35X
ATTEN3X
ATTEN3_5X
ATTEN400X
ATTEN40X
ATTEN450X
ATTEN45X
ATTEN4X
ATTEN4_5X
ATTEN500X
ATTEN50X
ATTEN550X
ATTEN55X
ATTEN5X
ATTEN5_5X
ATTEN600X
ATTEN60X
ATTEN650X
ATTEN65X
ATTEN6X
ATTEN6_5X
ATTEN700X
ATTEN70X
ATTEN750X
ATTEN75X
ATTEN7X
ATTEN7_5X
ATTEN800X
ATTEN80X
ATTEN850X
ATTEN85X
ATTEN8X
ATTEN8_5X
ATTEN900X
ATTEN90X
ATTEN950X
ATTEN95X
ATTEN9X
ATTEN9_5X
ATTempts
AUTHLogin
AUTHPassword
AUTO
AUTOAdjust
AUTOBright
AUTOSet
AUTOZero
AUX
AUXIn
AUXLevel
AUXiliary
AUXout
AVErage
B
B3ZS
B6ZS
B8ZS
BACKWards
BANdwidth
BASe
BAUd
BEEP
BELl
BETWeen
BINary
BITRate
BIT_Nr
BLACKANDWhite
BLACKMANHarris
BMP
BN_Fmt
BOTh
BOX
BOXPcnt
BTRIGger
BURst
BUSY
BY
BYPass
BYT_Nr
BYT_Or
CALProbe
CALibrate
CAN
CANH
CANL
CARea
CARet
CATalog
CENTER
CENTered
CH
CH1
CH14
CH2
CH3
CH4
CHECKSUMDR
CLAss
```

```
CLEAR
CLOCKCoax
CLOCKSymmetrical
CLOCk
CMDBatch
CMEan
CMI
CODe
COEFFS
COLOr
COMMunication
COMPLetion
COMpare
CONDition
CONSUmer
CONTROL
CONTROl
COPy
COUNT
COUNt
COUPling
CRMs
CROSSHair
CURSor
CURVENext
CURVEStream
CURVe
CUSTom
CWD
Custom
D1
D2
DASHed
DATA
DATE
DATa
DB
DBM
DC
DCREJect
DEFAULT
DEFAult
DEFine
DEGAUSS
DEGrees
DELEte
DELTATime
DELTa
DELay
DELayed
DESE
DESKEW
DESKew
DESTination
DIAg
DIFFerential
DIR
DIREction
DIRection
DISTDUty
DISable
DISplay
DIVisions
DOTs
DS0Contra
DS0Double
DS0Single
DS0Timing
DS1
DS1A
DS1C
DS2
DS2RATECoax
DS2RATESymmetrical
DS3
DS4NA
DS4NA_Max
E1
E1Coax
E1Symmetrical
E2
E3
E4
E4_0
E4_1
ECL
EDGE
EDGE2
EIGHtytwenty
EITher
EMail
ENCdg
ENET100
ENET1000BCX_ATP2
ENET1000BCX_ATP3
ENET1000BCX_NTP2
ENET100FX
ENET100STP
ENET100UTP
ENET1250
ENETXAUI
ENETXAUI_Far
ENETXAUI_Near
ENHANcedwfm
ENHanced
ENV
ENVelope
ENable
EOF
ET
EVEN
EVENT
EVENTS
EVMsg
EVQty
EXECUTE
EXECute
EXPort
EXTAtten
EXTDBatten
EXTENded
EXTINCTDB
EXTINCTPCT
EXTINCTRATIO
EXTUnits
EXTernal
EYE
EYEHeight
EYEWIdth
EYEdiagram
FACtory
FAILURES
FAILure
FALL
FALSe
FALling
FASTAcq
FASTERthan
FASTframe
FAStest
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_AD
FC2125E_ADR
FC2125E_ADT
FC2125E_AGR
FC2125E_AGT
FC2125E_NBT
FC2125E_NDT
FC2125E_NGT
FC266
FC266E
FC4250
FC4250E_ABR
FC4250E_ABT
FC4250E_ADR
FC4250E_ADT
FC4250E_AGR
FC4250E_AGT
FC4250E_NBT
FC4250E_NDT
FC4250E_NGT
FC531
FC531E
FFWD
FIELD
FIELD1
FIELD2
FIFtyfifty
FILE
FILEFormat
FILEName
FILESystem
FILTer
FILepath
FIVe
FLATTOP2
FORCEDRange
FORCe
FORMat
FORWards
FP
FPBinary
FRAMESTARt
FRAMESTOP
FRAMEStart
FRAMEtype
FRAMe
FRAme
FREQuency
FREV
FROm
FST
FST1
FST2
FST3
FST4
```

```
FST5
FST6
FULLNOmenu
FULLSCREEN
FULl
FUNCtion
FW1394BS1600B
FW1394BS1600BT1
FW1394BS1600BT2
FW1394BS400B
FW1394BS400BT1
FW1394BS400BT2
FW1394BS800B
FW1394BS800BT1
FW1394BS800BT2
FWD
FuLl
G703DS1
G703DS3
GAIN
GATEPOS
GATEWIDTH
GATing
GAUSSian
GENeralcall
GLItch
GND
GPIb
GRAticule
GRId
GROUPDelay
HALT
HAMMing
HANNing
HARDCopy
HBArs
HD1080I50
HD1080I60
HD1080P24
HD1080P25
HD1080SF24
HD480P60
HD720P60
HDB3
HDELTA
HDR
HEADer
HERtz
HEX
HFRej
HIGH
HIGHLIGHTHits
HIGHLimit
HIRes
HIStogram
HITS
HITs
HOLDTime
HOLDoff
HOLdoff
HORizontal
HORizontal[
HOSTwanted
HPOS
HSCAle
HST
HST1
HST2
HST3
HST4
HST5
HST6
HTRIGPOS
I2C
ID
IDANDDATA
IDENTifier
IMAGEView
IMAGe
IMMed
INDependent
INF2_5G
INF2_5GE
INFINIBAND
INFInite
INFPersist
INHERIT
INIT
INKSaver
INPUT
INPut
INSide
INTENSIFied
INTENSITy
INTERLAced
INTERNal
INTERPRatio
INTERnal
INVERTed
INVert
IRE
IT
ITEM
JPEG
KAISERBessel
LABel
LANdscape
LAST
LAYout
LENGTH
LENgth
LESSLimit
LESSThan
LEVEL
LEVel
LFRej
LIMit
LINE
LINEAr
LINES
LINESTyle
LINE_X
LIVE
LOCk
LOG
LOGIc
LONG
LOOP
LOOPS
LOW
LOWLimit
LOWerthreshold
LSB
LVLSrcpreference
MAG
MAIn
MAIn]
MANChester
MANual
MARgin
MASK
MASKHighlight
MASKPRE
MATH
MATH14
MATHArbflt
MATHCOLOr
MATHCad
MATHVAR
MATLab
MAXSamplerate
MAXSize
MAXimum
MEAN
MEANSTDdev
MEAS
MEASUrement
MEDian
METHod
MHZ10
MHZ100
MID
MINImum
MINMax
MINUSOne
MINimum
MISO
MKDir
MLT3
MODe
MONOGRAY
MONOGREEN
MORELimit
MOREThan
MOSI
MRTTime
MSB
MULTiscope
MULtipleframes
MV
NAMe
NANd
NCROss
NDUty
NEGAtive
NEWpass
NOCARE
NOISErej
NOISe
NONE
NONe
NOR
NORMALIZed
NORMal
NOVershoot
NR1
NR3
NRZ
NR_FR
NR_Pt
NTSC
NTSc
NUMACq
NUMAVg
NUMCOEFFS
NUMEMails
NUMERic
NUMEnv
NUMFRames
NUMITEMS
NUMSAMples
NWIdth
```

```
OC1
OC12
OC3
OC48
OC48_FEC
OCCurs
ODD
OFF
OFFGRAticule
OFFSETAdj
OFFSet
ON
ONE
ONEfifty
ONGRAticule
OPTIMnoise
OPTion
OR
ORR
OUT
OUTside
OVERlay
PAL
PALEtte
PARity
PASS
PASSWord
PATTERNBITS
PATtern
PBASe
PCIEXPRESS
PCIEXPRESS_Rcv
PCIEXPRESS_Xmit
PCIExpress
PCROss
PCTCROss
PCX
PDUty
PEAKHits
PEAKdetect
PERCent
PERIod
PERSistence
PHASE
PHAse
PK2Pk
PKPKJitter
PKPKNoise
PLOCK
PLOCK BITRate
PLUSOne
PNG
POINTS
POLARCoord
POLarity
PORT
PORTRait
POS
POSITION
POSITIVe
POSition
POVershoot
PRESAMPBITS
PRINTER
PRInt
PROBE
PROBECal
PROBEFunc
PROCON
PRODDELta
PRODUCT
PRODucer
PRODuct
PROGressive
PRObe
PRObestate
PTOP
PT_Fmt
PT_ORder
PT_Off
PULSEForm
PULse
PWIdth
QFACtor
QUAlify
RADIUS
RADians
RANDom
RANge
RATDELta
RATE1
RATE2
RATE3
RATE32Mbit
RATE4
RATE5
RATE97Mbit
RATIO
RATio
RDELta
READ
READFile
READOUT
READOuts
RECAll
RECORDView
RECOVered
RECOrdlength
RECTANGular
RECTX
RECTY
RECTangular
REF
REF14
REFCOLOr
REFLEVELOffset
REFLevel
REFOut
REJect
REM
REMote
REName
REPEATstart
REPeat
RESBw
RESET
RESETLive
RESOlution
RESULT
RESet
RESistance
RESults
REV
RI
RIBinary
RIO_1G
RIO_1_5G
RIO_2G
RIO_500M
RIO_750M
RIO_DRV1G
RIO_DRV1_5G
RIO_DRV2G
RIO_DRV500M
RIO_DRV750M
RIO_EDRV1G
RIO_EDRV1_5G
RIO_EDRV2G
RIO_EDRV500M
RIO_EDRV750M
RIO_RCV1G
RIO_RCV1_5G
RIO_RCV2G
RIO_RCV500M
RIO_RCV750M
RIO_SERIAL_1G
RIO_SERIAL_2G
RIO_SERIAL_3G
RISe
RISetime
RISing
RMDir
RMS
RMSJitter
RMSNoise
ROLL
ROLe
ROSc
RP
RPBinary
RS232
RT
RUN
RUNSTop
RUNT
S8B10B
SAMPLERate
SAMPlingmode
SAMple
SAS1_5
SAS1_5_CR
SAS1_5_CR_AASJ
SAS1_5_IR
SAS1_5_IR_AASJ
SAS1_5_SATA
SAS1_5_XR
SAS1_5_XR_AASJ
SAS3_
SAS3_0_CR
SAS3_0_CR_AASJ
SAS3_0_IR
SAS3_0_IR_AASJ
SAS3_0_SATA
SAS3_0_XR
SAS3_0_XR_AASJ
SAVEWFM
SAVe
SCAN
SCAle
SCLK
SCREEN
SCREENSAVER
SCREENSAVERDELAY
SCREENTExt
SCROLL
SCROLLLock
SECAM
SEConds
SEG
SELECTED
SELect
SEQuence
```

SERIAL
SERIALTRIG
SERnumber
SET
SETHold
SETLevel
SETTime
SETUp
SFI5_2
SFI5_3
SFI5_RCVBCLK2
SFI5_RCVBCLK3
SFI5_RCVBDATA2
SFI5_RCVBDATA3
SFI5_RCVDCLK2
SFI5_RCVDCLK3
SFI5_RCVDDATA2
SFI5_RCVDDATA3
SFI5_XMITACLK2
SFI5_XMITACLK3
SFI5_XMITADATA2
SFI5_XMITADATA3
SFI5_XMITCCLK2
SFI5_XMITCCLK3
SFI5_XMITCDATA2
SFI5_XMITCDATA3
SFPbinary
SHIFted
SHORt
SHOWREmote
SIGMA1
SIGMA2
SIGMA3
SIGType
SIGnal
SINX
SIXSigmajit
SIZE
SIZe
SLOWERthan
SLOpe
SMTPPort
SMTPServer
SNAp
SNRatio
SOF
SOLID
SOURCE
SOUrce
SPAN
SPC
SPECTral
SPEED
SPEed
SPI
SPLit
SPREADSHEETCsv
SPREADSHEETTxt
SRCDependent
SRCIndependent
SRIbinary
SRPbinary
SRQ
SS
STANDARD
STANdard
STARt
STARtbyte
STATE
STATIstics
STATUS
STATus
STAYSHigh
STAYSLow
STDdev
STM0_0
STM0_1
STM0_CMI
STM0_HDBX
STM1E
STM1E_0
STM1E_1
STOP
STOPAfter
STOPOnviolation
STORe
STRing
STS1
STS1Eye
STS1Pulse
STS3
STS3_Max
STYle
SUBITEMS
SUBSYS
SUMFrame
SUPPress
TEKEXPonential
TEKSecure
TEMPErature
TEMPLate
TEMPlate
TENbit
TERmination
TEST
TESTS
TESt
TFI5_2
TFI5_3
THDELta
THETA
THReshold
TIFF
TIME
TIMEOut
TIMEPosition
TIMEStamp
TIMe
TO
TOLerance
TOTal
TRACk
TRANsition
TRIG
TRIGBar
TRIGIF
TRIGT
TRIGTOSAMP
TRIGger
TRIgger
TRUe
TTL
TWEnty
TWOfifty
TYPe
UNDEFINED
UNDo
UNITString
UNIts
UNLock
UNWRap
UPPerthreshold
USER
USERMask
USEr
VALUEMean
VALue
VAR
VARpersist
VBArs
VDELTA
VECtors
VERBose
VERT_INDEX
VERTical
VFields
VIDEO270
VIDEO360
VIDeo
VIEW
VIOLATIONS
VLines
VOFFSet
VPOS
VSCAle
VSROC192
WAVEFORMS
WAVEform
WAVFRMStream
WAVFrm
WEIghting
WFId
WFMDB
WFMInpre
WFMOutpre
WFMPre
WHEn
WIDERthan
WIDTH
WIDth
WINdow
WIThin
WRITE
WRITEFile
X
XDELta
XINcr
XPOS
XPOSITION
XUNit
XY
XZEro
Y
YDELta
YMUlt
YOFf
YPOS
YPOSITION
YT
YUNit
YZEro
ZERO
ZERo
ZOOM
ZOOm

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 CH1;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)

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	SOLID

Command	Default Values
CURSor:MODe	INDEPENDENT
CURSor:SCREEN:STYle	LINES
CURSor:SOUrce<x>	CH1
CURSor:STATE	0
CURSor:VBArs:POSITION<x>	1 = -1.6000E-6 2 = 1.6000E-6
CURSor:VBArs:UNIts	SECONDS
CURSor:WAVEform:POSition<x>	1 = -1.6000E-6 2 = 1.6000E-6
CURSor:WAVEform:SOUrce<x>	1 = CH1 2 = CH1
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

Display 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	TEMPERATURE
DISplay:COLOr:PALEtte:RECORDView	NORMAL
DISplay:COLOr:PALEtte:USEr:CARet	150,50,100
DISplay:COLOr:PALEtte:USEr:CH<x>	1 = 180,50,100 2 = 300,50,100 3 = 60,50,100 4 = 240,50,100
DISplay:COLOr:PALEtte:USEr:GRAticule	165,50,15
DISplay:COLOr:PALEtte:USEr:HIStogram	320,50,100
DISplay:COLOr:PALEtte:USEr:MASK	0,25,75
DISplay:COLOr:PALEtte:USEr:MASKHighlight	140,50,100

Command	Default Values
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 = 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:PERSistence	OFF
DISplay:SCREENTExt:LABel<x>:NAMe	""
DISplay:SCREENTExt:LABel<x>:XPOS	100
DISplay:SCREENTExt:LABel<x>:YPOS	1 = 5 2 = 20 3 = 35 4 = 50 5 = 65 6 = 80 7 = 95 8 = 110
DISplay:SCREENTExt:STATE	0
DISplay:STYle	VECTORS
DISplay:TRIGBar	SHORT
DISplay:TRIGT	1
DISplay:VARpersist	500.0000E-3

Email Default Values. The following table lists the Email setup values.

Command	Default Values
EMail:ATTempts	1
EMail:AUTHLogin	""
EMail:FROm	"DPO7104_<Instrument Serial Number>"
EMail:HOSTwanted	""
EMail:IMAGe	0

Command	Default Values
EMail:MASK	0
EMail:MAXSize	1
EMail:MEASUrement	0
EMail:NUMEMails	1
EMail:SMTPPort	25
EMail:SMTPServer	""
EMail:TIMEOut	30
EMail:TO	""
EMail:TRIGger	0
EMail:WAVEform	0

Hardcopy Default Values. The following table lists the Hardcopy factory default setup values.

Command	Default Values
HARDCopy:FILEName	""
HARDCopy:PORT	FILE

Histogram Default Values. The following table lists the Histogram factory default setup values.

Command	Default Values
HIStogram:BOXPcnt	30.0000,25.1000,70.0000,75.2000
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[:MAIn]:DELay:MODe	0
HORizontal[:MAIn]:DELay:POSition	50.0000
HORizontal[:MAIn]:DELay:TIMe	0.00000000000
HORizontal[:MAIn]:POSition	50.0000
HORizontal:MAIn:SAMPLERate	≥2.5 GHz instruments: 250E+6 ≤2.5 GHz instruments: 200E+6
HORizontal[:MAIn]:SCAle	≥2.5 GHz instruments: 400.0000E-9 ≤2.5 GHz instruments: 500.0000E-9

Command	Default Values
HORizontal:RECOrdlength	1000
HORizontal:RESOlution	5000
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
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

Command	Default Values
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
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	" "

Command	Default Values
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
MATH<x>:SPECTral:MAG	DB
MATH<x>:SPECTral:PHASE	DEGREES
MATH<x>:SPECTral:REFLevelM	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:SUPPress	-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

Command	Default Values
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
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:SOUrce	INTERNAL
VERBose	1 (ON)

Save and Recall Default Settings. The following table lists the Save and Recall factory default setup values.

Command	Default Values
SAVe:WAVEform:FILEFormat	INTERNAL

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
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-9
TRIGger:{A|B}:LOGIc:PATtern:WHEn:MORELimit	5.0000E-9
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

Command	Default Values
TRIGger:{A|B}:LOGIc:SETHold:SETTime	3.0000E-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
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:DELTATime	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	FASTERTHAN
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
	INSIDE
	OCCURS
TRIGger:{A|B}:PULse:WINdow:WIDTH	2.0000E-9
TRIGger:A:TYPe	EDGE
TRIGger:A:VIDeo:CUSTom:FORMat	INTERLACED
	RATE1
TRIGger:A:VIDeo:FIELD	ALLFIELDS
TRIGger:A:VIDeo:HOLdoff:FIELD	0.0000
TRIGger:A:VIDeo:LINE	1
TRIGger:A:VIDeo:POLarity	NORMAL

Command	Default Values
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
TRIGger:B:TYPe	EDGE

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
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>: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.

Command	Default Values
DATa:DESTination	REF1
DATa:ENCdg	RIBINARY
DATa:SOUrce	CH1
DATa:STARt	1
DATa:STOP	5000

Zoom Default Values. The following table lists the Zoom factory default setup values.

Command	Default Values
ZOOm:GRAticule:SIZE	80
ZOOm:MODe	0
ZOOm:SCROLL:DIREction	STOP
ZOOm:SCROLL:LOCk	0
ZOOm:SCROLL:SPEED	1
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:ZOOM<x>:STATE	1

Appendix D: GPIB Interface Specifications

GPIB Interface Specifications

Overview. 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

Message	Interface
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.

Table D-2: Interface Functions, (cont.)

Interface Function	Subset	Meaning
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 POll 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