Tektronix®

TDS7000 Series
GPIB Programmer Online Guide

Printable version 1.0

Copyright and Version Information

Tektronix, Inc. P.O. Box 500, Beaverton, OR 97077

TDS7000 Series Programmer Online Guide

OLH0140, Version 01.00

Getting Started

Introduction

This programmer online guide provides you with the information you need 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.

The programmer guide is divided into the following major topics (books):

- **Getting Started.** This topic introduces you to the online help and provides basic information about setting up your instrument for remote control.
- **Command Syntax.** This topic provides an overview of the command syntax that you will use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command Groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing and a detailed description of the command will be displayed.
- **Status and Events.** This topic discusses the status and event reporting system for the GPIB interface. 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 GPIB interface specifications that may be helpful when using GPIB commands to remotely control the instrument.

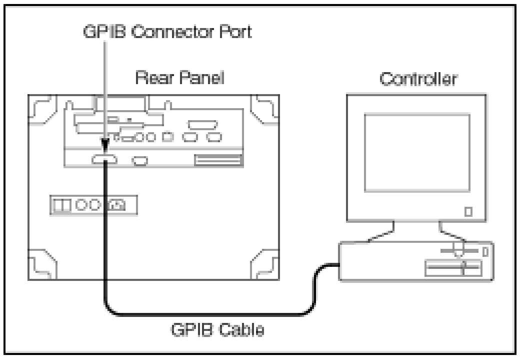
Setting Up Remote Communications

Before setting up your instrument for remote communications, you should familiarize yourself with the following GPIB requirements:

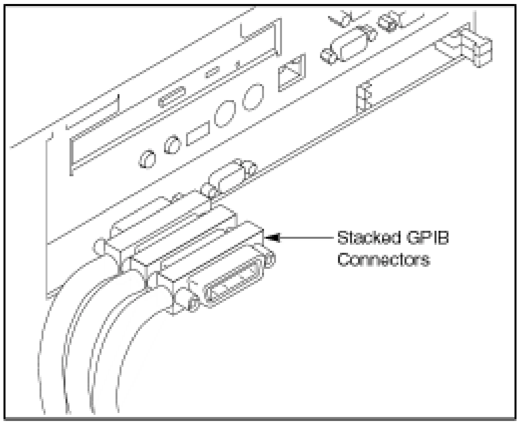
- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No more than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No more than 65 feet (20 meters) of cable should 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 panel. This connector has a D-type shell and conforms to IEEE Std 488.1–1987. Attach an IEEE Std 488.1–1987 GPIB cable to this connector and to your controller as shown in the following figure.



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



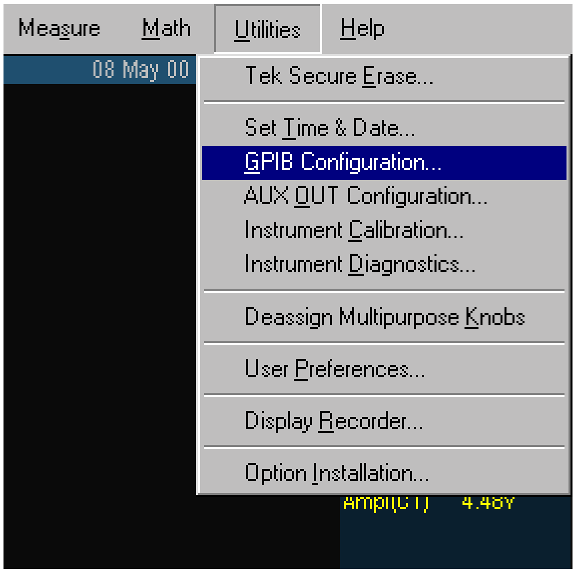
Setting the GPIB Address

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

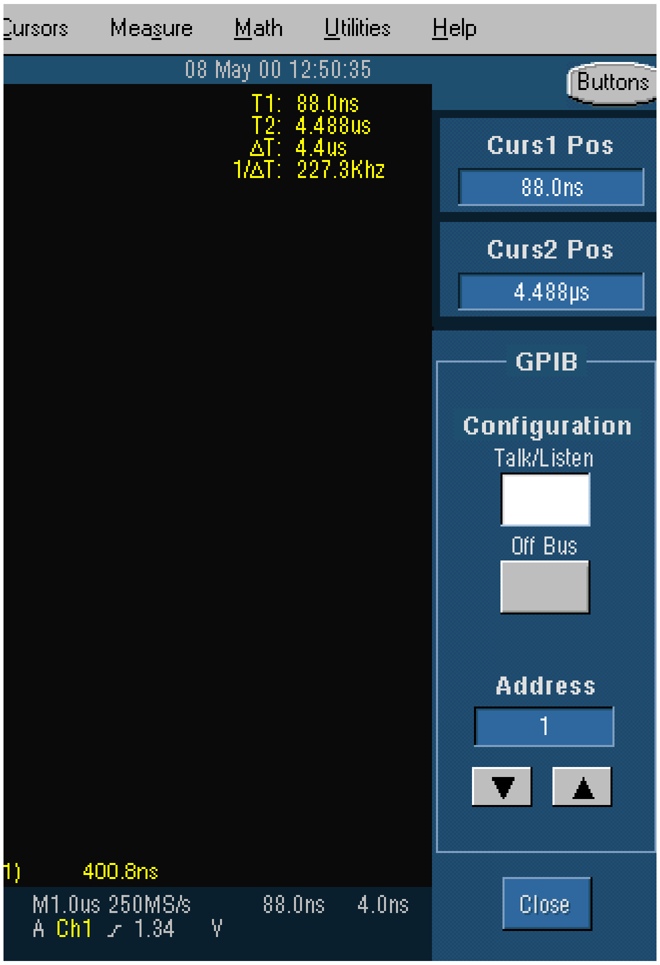
• GPIB Address	1
• GPIB Mode	GPIB Talk/Listen

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.



```
08 May 00 12:50:35
T1:  88.0ns
T2:  4.488us
ΔT:  4.4us
1/ΔT:  227.3Khz
1)        400.8ns
M1.0us 250MS/s        88.0ns    4.0ns
A Ch1 ⟋ 1.34     V
```

Curs1 Pos: 88.0ns

Curs2 Pos: 4.488µs

Address: 1

3. Change the GPIB Address to a unique address.

4. Click the Close button.

The instrument is set up for bi-directional communication with your controller.

Documentation

In addition to this *TDS7000 Series Programmer Online Guide*, the following documentation is included with this instrument.

***User Manual*.** The User Manual has information about installing and operating the Oscilloscope. It also provides concepts and theories about using the oscilloscope that are not covered in the online help.

***Quick Reference Guide*.** A quick reference guide provides basic information on connecting a signal to the oscilloscope and displaying the results.

***TDS 7000 Series Online Help*.** This is an online help system that is integrated with the User Interface application that ships with this product.

***Optional Service Manual*.** A service manual is available as an optional accessory; it includes procedures to service the oscilloscope to module levels. It also includes performance verification procedures so you can verify performance to advertised specifications.

Command Syntax

Syntax Overview

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.

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.

Command and 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, restrictijon, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

Commands

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

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

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

Queries

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

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

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level. For example, HIStogram:STATistics:STDdev? returns the standard deviation of the histogram, while HIStogram:STATistics? returns all the histogram statistics, and HIStogram? returns all the histogram parameters.

Headers

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

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 Entry

Rules

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 background color is white and the display foreground color is black, the concatenated query DISplay:COLor:BACKGround?;FOREGround? will return the following.

If the header is on:

```
:DISPLAY:COLOR:BACKGROUND 7;:DISPLAY:COLOR:FOREGROUND 0
```

If the header is off:

```
7;0
```

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.

End of Message Terminator	
Symbol	**Meaning**
<EOM>	Message terminator

The end-of-message terminator may be the END message (EOI asserted concurrently with the last data byte), the ASCII code for line feed (LF) sent as the last data byte, or both. The instrument always terminates messages with LF and EOI. It allows white space before the terminator. For example, it allows CR LF.

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.

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:

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

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

Quoted String

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

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 simply 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 1000 characters.

Here are some invalid strings:

* "Invalid string argument' (quotes are not of the same type)

* "test<EOI>" (termination character is embedded in the string)

Block

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

Block Arguments

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 <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

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.

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.

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.

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.

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.

Reference Waveform Mnemonics	
Symbol	**Meaning**
REF<x>	A reference waveform specifier; <x> is 1 through 4.

Commands

Acquisition Command Group

Acquisition Overview

You 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.

Acquisition Commands

Command	Description
ACQuire?	Returns acquisition parameters
ACQuire:MODe?	Returns acquisition mode
ACQuire:MODe	Sets acquisition mode
ACQuire:NUMACq?	Returns the number of acquisitions that have occurred
ACQuire:NUMAVg?	Returns number of acquisitions for an averaged waveform
ACQuire:NUMAVg	Sets number of acquisitions for an averaged waveform
ACQuireNUMEnv?	Returns number of acquisitions for envelope waveform
ACQuire:NUMEnv	Sets number of acquisitions for envelope waveform
ACQuire:REPEt?	Returns whether repetitive signal acquisition mode on
ACQuire:REPEt	Enables or disables repetitive signal acquisition mode
ACQuire:STATE?	Returns acquisition state
ACQuire:STATE	Starts or stops acquisitions
ACQuire:STOPAfter?	Returns whether the acquisition is continuous or single sequence
ACQuire:STOPAfter	Sets whether the acquisition is continuous or single sequence
FASTAcq?	Returns the Fast Acquisition state
FASTAcq:STATE?	Returns the Fast Acquisition state
FASTAcq:STATE	Enables or disables Fast Acquisition mode

ACQuire:MODe

Description

This command sets or queries the acquisition mode of the oscilloscope. 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 oscilloscope 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

Related Commands

ACQuire:NUMAVg (see page 23), ACQuire:NUMenv (see page 24), CURVe? (see page 467)

Syntax 1

```
ACQuire:MODe{SAMple|PEAKdetect|HIRes|AVErage|ENVelope}
```

Syntax 2

```
ACQuire:MODe?
```

Arguments

- `SAMple`

  Specifies that the displayed data point value is simply 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. For more information about the CURVe query, see page 467.

- `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 oscilloscope 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. For more information about the ACQuire:NUMACg command, see page 23.

- `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. For more information about the ACQuire:NUMENv command, see page 24.

Example 1

```
ACQuire:MODe ENVelope
```

Sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

Example 2

```
ACQuire:MODe?
```

This command might return `:ACQuire:MODe AVERAGE`, indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

ACQuire:NUMACq?

Description

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^{63}–1. The oscilloscope stops counting when this number is reached. This is the same value that displays in the upper center of the screen when the acquisition system is stopped.

Note: In FastFrame or InstaVu modes, the oscilloscope updates the acquisition count in groups of counts.

Group

Acquisition

Related Commands

ACQuire:STATE (see page 26)

Syntax

```
ACQuire:NUMACq?
```

Example

```
ACQuire:NUMACq?
```

This command might return `:ACQUIRE:NUMACQ 350`, indicating that 350 acquisitions have occurred since executing the ACQuire:STATE RUN command.

ACQuire:NUMAVg

Description

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 then 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

Related Commands

ACQuire:MODe (see page 20)

Syntax 1

```
ACQuire:NUMAVg <NRf>
```

Syntax 2

```
ACQuire:NUMAVg?
```

Arguments

- `NRf`

  This is the number of waveform acquisitions to average.

Example 1

```
ACQuire:NUMAVg 10
```

This command specifies that 10 waveform averages will be performed before exponential averaging starts.

Example 2

```
ACQuire:NUMAVg?
```

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

ACQuire:NUMEnv

Description

This command sets or queries the number of waveform acquisitions that make up an envelope waveform. Sending this command is equivalent to setting the Envelope count in the Acquisition Mode side menu.

Group

Acquisition

Syntax 1

```
ACQuire:NUMEnv {<NRf> | INFInite}
```

Syntax 2

```
ACQuire:NUMEnv?
```

Arguments

- `NRf`

  This is the number of waveform acquisitions to be enveloped.

Example 1

```
ACQuire:NUMenv 10
```

This command specifies that an enveloped waveform will show the result of combining 10 separately acquired waveforms.

Example 2

```
ACQuire:NUMenv?
```

This command might return `:ACQUIRE:NUMENV 0`, indicating that acquisitions are acquired infinitely for enveloped waveforms.

ACQuire:REPEt

Description

This command sets or queries whether repetitive mode is on or off. This is equivalent to setting Equivalent Time Auto/Off in the Acquisition control window. When the oscilloscope is in real-time operation, this setting has no effect.

The ACQuire:REPet command specifies the behavior of the acquisition system during equivalent-time operation. When repetitive mode is on, the acquisition system will continue to acquire waveform data until the waveform record is filled with acquired data. When repetitive mode is off, horizontal scale and record length settings will be restricted to those settings that will be achievable with real-time acquisition.

Group

Acquisition

Syntax 1

```
ACQuire:REPEt  {OFF|ON|<NR1>}
```

Syntax 2

```
ACQuire:REPEt?
```

Arguments

- `OFF`

  This argument disables repetitive mode.

- `ON`

  This argument enables repetitive mode.

- `NR1`

  A 0 disables repetitive mode ; any other value enables repetitive mode.

Example 1

```
ACQUIRE:REPEt  1
```

This command enables repetitive mode.

Example 2

```
ACQuire:REPEt?
```

This query might return `:ACQUIRE:REPET  OFF`, indicating that repetitive mode is disabled.

ACQuire:STATE

Description

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

Sending this command is equivalent to pressing the front-panel RUN/STOP button.

Group

Acquisition

Related Commands

ACQuire:STOPAfter (see page 27)

Syntax 1

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

Syntax 2

```
ACQuire:STATE?
```

Arguments

- `OFF`

  This argument stops acquisitions.

- `STOP`

  This argument stops acquisitions.

- `ON`

  This argument starts acquisitions.

- `RUN`

  This argument starts acquisitions.

- `NR1`

  0 stops acquisitions; any other value starts acquisitions.

Example 1

```
ACQuire:STATE RUN
```

This command starts the acquisition of waveform data and resets the count of the number of acquisitions.

Example 2

```
ACQuire:STATE?
```

This query might return `:ACQUIRE:STATE  0`, indicating that the acquisition is stopped

ACQuire:STOPAfter

Description

This command sets or queries whether the oscilloscope continually acquires acquisitions or acquires a single sequence. This command is equivalent to pressing **SINGLE** from the front-panel.

Group

Acquisition

Related Commands

ACQuire:STATE (see page 26), ACQuire:REPEt (see page 25)

Syntax 1

```
ACQuire:STOPAfter  {RUNSTop|SEQuence}
```

Syntax 2

```
ACQuire:STOPAfter?
```

Arguments

- `RUNSTop`

  Specifies that the oscilloscope will continually acquire data, if ACQuire:STATE is turned on.

- `SEQuence`

  Specifies that the next acquisition will be a single-sequence acquisition.

Example 1

```
ACQuire:STOPAfter RUNSTOP
```

This command sets the oscilloscope to continually acquire data.

Example 2

```
ACQuire:STOPAfter?
```

This query might return `:ACQUIRE:STOPAFTER SEQUENCE`, indicating that the next acquisition the oscilloscope makes will be of the single-sequence type.

ACQuire?

Description

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

- Stop after
- Acquisition state
- Mode
- Number of envelopes
- Number of averages
- Repetitive signals

Group

Acquisition

Related Commands

ACQuire:MODe (see page 20), ACQuire:NUMACq (see page 22), ACQuire:NUMAVg (see page 23), ACQuire:NUMenv (see page 24), ACQuire:REPEt (see page 25), ACQuire:STATE (see page 26), ACQuire:STOPAfter (see page 27)

Syntax

```
ACQuire?
```

Example

```
ACQuire?
```

This query might return the following string for the current acquisition parameters:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET ON
```

FASTAcq?

Description

This query-only command returns the state of Fast Acquisition. This command is equivalent to the **FASTACQ** button on the front panel.

Group

Acquisition

Syntax

```
FASTAcq?
```

Example

```
FASTAcq?
```

This query might return the following string for the current acquisition parameters:

```
:FASTACQ:STATE 0
```

FASTAcq:STATE

Description

This command sets or queries the state of Fast Acquisition. This command is equivalent to the **FASTACQ** button on the front panel.

Group

Acquisition

Syntax 1

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

Syntax 2

```
FASTAcq:STATE?
```

Arguments

- `ON`

  This enables Fast Acquisition mode.

- `OFF`

  This disables Fast Acquisition mode.

- `<NR1>`

  A 0 disables Fast Acquisition mode; any other value enables Fast Acquisition mode.

Example 1

```
FASTAcq:STATE ON
```

This command enables the FastAcq operation.

Example 2

This query might return `:FASTACQ:STATE 1`, indicating that the Fast Acquisition state is currently active.

Alias Command Group

Alias Overview

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 Phoenix errors unless specifically noted in the command description in this document.

Alias Commands

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

ALIas

Description

This command sets or queries the state of alias functionality.

Group

Alias

Related Commands

ALIas:STATE (see page 39)

Syntax 1

```
ALIas {OFF|ON|<NR1>}
```

Syntax 2

```
ALIas?
```

Arguments

- `OFF`

  This turns Alias expansion off. If a defined alias is sent when ALIas:STATE is off, a command error (102) will be generated.

- `ON`

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

- `NR1`

  A 0 disables Alias mode; any other value enables Alias mode.

Example 1

```
ALIas ON
```

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

Example 2

```
ALIas?
```

This query might return the following string for the current alias state:

```
:ALIAS:STATE 1
```

ALIas:CATalog?

Description

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?
```

Example

```
ALIas:CATalog?
```

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

ALIas:DEFIne

Description

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

Related Commands

ALIas:STATE (see page 39)

Syntax 1

```
ALIas:DEFIne <Qstring><,>{<Qstring>|<Block>}
```

Syntax 2

```
ALIas:DEFIne?
```

Arguments

- `The first <Qstring> is the alias label.`

  This label cannot be a command name. Labels must start with a letter and can contain 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.

Example 1

```
ALIas:DEFIne "ST1",":RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON"
```

This command defines an alias named "ST1" that sets up the oscilloscope.

Example 2

```
ALIas:DEFIne? "ST1"
```

This command returns `:ALIAS:DEFINE "ST1",#246`
`:RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON`

ALIas:DELEte

Description

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

Group

Alias

Related Commands

*ESR? (see page 36), ALIas:DELEte:ALL (see page 37)

Syntax

```
ALIas:DELEte <Qstring>
```

Argument

- `<Qstring>`

  This 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.

Example

```
ALIas:DELEte "SETUP1"
```

This command deletes the alias named SETUP1.

ALIas:DELEte:ALL

Description

This command deletes all existing aliases.

Group

Alias

Related Commands

ALIas:DELEte (see page 36), ALIas:DELEte:NAMe (see page 38)

Syntax

```
ALIas:DELEte:ALL
```

Example

```
ALIas:DELEte:ALL
```

This command deletes all existing aliases.

ALIas:DELEte:NAMe

Description

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>
```

Argument

- `<Qstring>`

  This 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.

Example

```
ALIas:DELEte:NAMe "STARTUP"
```

This command deletes the alias named STARTUP.

ALIas:STATE

Description

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 1

```
ALIas:STATE {OFF|ON|<NR1>}
```

Syntax 2

```
ALIas:STATE?
```

Arguments

- `OFF`

  This turns alias expansion off. If a defined alias is sent when ALIas:STATE is OFF, the instrument generates a command error (102).

- `ON`

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

- `<NR1>`

  A 0 turns off aliases; any other value turns on aliases.

Example 1

```
ALIas:STATE OFF
```

This command turns off the alias feature.

Example 2

```
ALIas:STATE?
```

This query might return `:ALIAS:STATE ON`, indicating that alias expansion is currently turned on.

Calibration Command Group

Calibration Overview

The calibration commands provide information about the current state of calibration for the mainframe and all resident sampling head channels.

Calibration Commands

Command	Description
*CAL?	Instructs the scope to perform self-calibration
CALibrate:PRObestate:CH<x>?	Returns the calibration status of the attached probe
CALibrate:RESults?	Returns status of all calibration subsystems without performing an SPC operation
CALibrate:RESults:SPC?	Returns the results of the last SPC operation

*CAL?

Description

This query-only command instructs the digitizing oscilloscope to perform a signal path compensation and returns its calibration status. This command is equivalent to selecting Instrument Calibration from the Utilities menu and then clicking the Calibrate button.

> **Note:** The self-calibration can take several minutes to respond. No other commands will be executed until calibration is complete.

Group

Calibration

Related Commands

CALibrate:RESults_SPC? (see page 44)

Syntax

```
*CAL?
```

Example

```
*CAL?
```

This query performs an internal self-calibration that returns `*CAL 0` to indicate that the calibration was successful or nonzero to indicate a failure.

CALibrate:RESults?

Description

This query-only command returns the status of all the calibration subsystems, without performing an SPC operation. The results returned do not include the calibration status of attached probes. The CALibration:RESults? Query is intended to support GO/NoGO testing of the oscilloscope's calibration readiness: all returned results should indicate Pass status if the oscilloscope is "fit for duty". It is quite common, however, to use uncalibrated probes (particularly when the oscilloscope inputs are connected into a test system with coax cables). Not including the probe cal results prompts the instrument to perform a simple "every returned status field must be Pass" test.

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

Group

Calibration

Related Commands

*CAL? (see page 41)

Syntax

```
CALibrate:RESults?
```

Example

```
CALibrate:RESults?
```

This query returns the status of all the calibration subsystems. The query might return

```
:CALIBRATE:RESULTS:SPC PASS;VOLTAGE PASS;FREQUENCY PASS;TRIGGER PASS
```

CALibrate:PRObestate:CH<x>?

Description

This query-only command returns the status of probe calibration of a given channel's probe, which can be Pass (0), Initialized, (1), Fail (-1), or Running (2). This command is equivalent to selecting Probe Cal from the Vertical menu.

Group

Calibration

Related Commands

CH<x>:PROBECal? (see page 439)

Syntax

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

Example

```
CALibrate:PRObestate:CH1?
```

This query returns the status of the probe calibration of a given channel's probe. A query might return `:CALIBRATE:PROBESTATE:CH1 0`, indicating that the probe calibration passed.

CALibrate:RESults:SPC?

Description

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

Related Commands

*CAL? (see page 41)

Syntax

```
CALibrate:RESults:SPC?
```

Arguments

None.

Example

```
CALibrate:RESults:SPC?
```

This query returns the results of the last SPC operation.

Cursor Command Group

Cursor Overview

You 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.
- Paired Cursors—displays measurement of horizontal (time) and vertical (voltage) difference between paired cursors; voltage at the vertical position of the specified paired cursor; horizontal position of the specified paired cursor; and measurement units for the paired cursors.
- Split Cursors—displays measurement of horizontal (time) and vertical (voltage) difference between split cursors; voltage at the vertical position of the specified split cursor; horizontal position of the specified split cursor; and measurement units for the split cursors.

Cursor Commands

Command	Description
CURSor?	Returns all cursor settings
CURSor:FUNCtion?	Returns cursor type
CURSor:FUNCtion	Sets cursor type
CURSor:HBArs?	Returns hbar cursor settings
CURSor:HBArs:DELTa?	Returns hbars vertical difference
CURSor:HBArs:POSITION<x>?	Returns hbar cursor<x> vertical position
CURSor:HBArs:POSITION<x>	Sets hbar cursor<x> vertical position
CURSor:HBArs:UNIts?	Returns hbar cursor units
CURSor:MODe?	Returns whether cursors move in unison or separately
CURSor:MODe	Sets whether cursors move in unison or separately
CURSor:PAIred?	Returns paired cursor positions
CURSor:PAIred	Positions active paired cursor
CURSor:PAIred:HDELTA?	Returns vertical difference between two paired cursors
CURSor PAIred:HPOS<x>?	Returns the voltage at the vertical position of the specified paired cursor
CURSor:PAIred:POSITION<x>?	Returns the horizontal position of the specified paired cursor
CURSor:PAIred:POSITION<x>	Sets the horizontal position to the specified paired cursor
CURSor:PAIred:UNIts?	Returns the units for the paired cursors
CURSor:PAIred:VDELTA?	Returns the vbar difference between paired cursors
CURSor:SOUrce?	Returns the source for cursors
CURSor:SOUrce	Sets the source for cursors
CURSor:SPLit?	Returns the units, position, and second source of the split cursors
CURSor:SPLit	Sets the split cursors positions
CURSor:SPLit:HDELTA?	Returns the vertical difference between cursors 1 and 2
CURSor:SPLit:HPOS<x>?	Returns the vertical position of cursor 1 or 2
CURSor:SPLit:POSITION<x>	Sets the horizontal position of the vbar markers for cursor 1 or 2
CURSor:SPLit:POSITION<x>?	Returns the horizontal position of the vbar markers for cursor 1 or 2
CURSor:SPLit:SOURCE2?	Returns the source waveforms for split cursor 2
CURSor:SPLit:SOURCE2	Sets the source waveforms for split cursor 2
CURSor:SPLit:UNIts?	Returns the amplitude units of the split

	cursors
CURSor:SPLit:UNIts	Sets the amplitude units of the split cursors
CURSor:SPLit:VDELTA?	Returns the horizontal difference between cursors 1 and 2
CURSor:STATE?	Returns the cursor state
CURSor:STATE	Turns cursors on or off
CURSor:VBArs?	Returns the position of the vertical bar cursor
CURSor:VBArs	Sets the position of the vertical bar cursor
CURSor:VBArs:DELTa?	Returns difference between vbar cursors
CURSor:VBArs:POSITION<x>?	Returns vbar and paired cursor<x> positions
CURSor:VBArs:POSITION<x>	Sets vbar and paired cursor<x> positions
CURSor:VBArs:UNIts?	Returns the units for the vbar cursors
CURSor:VBArs:UNIts	Sets the units for the vbar cursors

CURSor?

Description

This query-only command returns all of the current cursor settings.

Group

Cursor

Syntax

```
CURSor?
```

Example: `CURSOR?`

This query might return the following as the current cursor settings:

```
:CURSOR:STATE OFF;FUNCTION VBARS;MODE INDEPENDENT
;SOURCE CH1;VBARS:UNITS SECONDS;POSITION1 -8.0000E-07
;POSITION2 8.0000E-07;:CURSOR:HBARS:POSITION1 3.0000E-01
;POSITION2 -3.0000E-01;UNITS BASE;:CURSOR:PAIRED
:POSITION1 -8.0000E-07;POSITION2 8.0000E-07;UNITS BASE;
:CURSOR:SPLIT:POSITION1 -8.0000E-07;POSITION2 8.0000E-07
;SOURCE2 CH1;UNITS BASE
```

CURSor:FUNCtion

Description

This command selects or queries the cursor type. Sending this command is equivalent to selecting Cursor Type from the Cursor menu and then choosing from the drop-down list.

Group

Cursor

Related Commands

CURSor:STAte (see page 69)

Syntax 1

```
CURSor:FUNCtion {HBArs|OFF|VBArs|PAIred|SPLit}
```

Syntax 2

```
CURSor:FUNCtion?
```

Arguments

- `HBArs`

  Specifies horizontal bar cursors, which measure in vertical units.

- `OFF`

  Removes the cursors from the display but does not change the cursor type.

- `VBArs`

  Specifies vertical bar cursors, which measure in horizontal units.

- `PAIred`

  Specifies paired cursors that measure in both horizontal and vertical units.

- `SPLit`

  Specifies split cursors, which measure horizontal and vertical units across two waveforms.

Example 1

```
CURSOR:FUNCtion VBARS
```

This command selects vertical bar type cursors.

Example 2

```
CURSOR:FUNCtion?
```

This query might return `:CURSOR:FUNCTION HBARS`, indicating that the horizontal bar cursors are currently selected.

CURSor:HBArs?

Description

This query-only command returns the current settings for the horizontal bar cursors.

Group

Cursor

Syntax

```
CURSor:HBArs?
```

Example

```
CURSor:HBArs?
```

This command might return `:CURSOR:HBARS:POSITION1 320.0000E-03;POSITION2-320.0000E-03;UNITS BASE`

CURSor:HBArs:DELTa?

Description

This query-only command returns the vertical difference between the two horizontal bar cursors.

Group

Cursor

Syntax

```
CURSor:HBArs:DELTa?
```

Returns

<NR3>

Example

```
CURSOR:HBArs:DELTa?
```

This command 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>

Description

This command specifies or queries the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). This command is the equivalent of 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 1

```
CURSor:HBArs:POSITION<x> <NRf>
```

Syntax 2

```
CURSor:HBArs:POSITION<x>?
```

Argument

- `<NR3>`

  Specifies the cursor position relative to ground.

Example 1

```
CURSOR:HBARS:POSITION1 25.0E-3
```

Positions one of the horizontal cursors at 25 mV.

Example 2

```
CURSOR:HBARS:POSITION2?
```

This query might return `:CURSOR:HBARS:POSITION2 -64.0000E-03`, indicating that one of the horizontal bar cursors is at -64 mV.

CURSor:HBArs:UNIts?

Description

This query-only command returns the units for the horizontal bar cursors.

Group

Cursor

Syntax

```
CURSor:HBArs:UNIts?
```

Example

```
CURSor:HBArs:UNIts?
```

This query might return `:CURSOR:HBARS:UNITS BASE` indicating that the units for the horizontal bar cursors are base.

CURSor:MODe

Description

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 Cursor menu and then choosing from the drop-down list.

Group

Cursor

Syntax 1

```
CURSor:MODe {TRACk|INDependent}
```

Syntax 2

```
CURSor:MODe?
```

Arguments

- `TRACk`

  This ties the navigational functionality of these 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`

  This frees the two cursors to be independently adjusted.

Example 1

```
CURsor:MODe TRACk
```

Specifies that the cursor positions move in unison.

Example 2

```
CURSor:MODe?
```

This query might return `:CURSOR:MODE TRACK` indicating that the two cursors move in unison.

CURSor:PAIred

Description

This command positions the active paired cursor to the position defined by the DATa:STARt or DATa:STOP command. This query returns the current paired cursor settings.

Group

Cursor

Related Commands

DATa:STARt (see page 473), DATa:STOP (see page 474)

Syntax 1

```
CURSor:PAIred SNAp
```

Syntax 2

```
CURSor:PAIred?
```

Arguments

- `SNAp`

  This positions the paired cursors at DATa:STARt and DATa:STOP.

Returns

A string for horizontal units and two floating point values with an exponent for cursor1 position and cursor2 position.

Example 1

```
CURSor:PAIred SNAp
```

Specifies the positions of the cursors at the current DATa:STARt and DATa:STOP values.

Example 2

```
CURSor:PAIred?
```

This query might return `:CURSOR:PAIRED:POSITION1 -2.0000E-03;POSITION2 2.0000E-03;UNITS BASE`

CURSor:PAIred:HDELTA?

Description

This query-only command returns the vertical difference between the cursor1 and cursor2 paired cursors. This is the absolute value of the vertical position for cursor1 minus the vertical position for cursor2.

Group

Cursor

Related Commands

CURSor:SPLit:HDELTA? (see page 63)

Syntax

```
CURSor:PAIred:HDELTA?
```

Returns

<NR3>

Example

```
CURSor:PAIred:HDELTA?
```

This query might return `:CURSOR:PAIRED:HDELTA 5.0800E+00`, indicating that the voltage difference between the two cursors is 5.08 V.

CURSor:PAIred:HPOS<x>?

Description

This query-only command returns the voltage at the vertical position of the cursor<x> paired cursor. The paired cursor can be either 1 or 2.

Group

Cursor

Related Commands

CURSor:FUNCtion (see page 49)

Syntax

```
CURSor:PAIred:HPOS<x>?
```

Returns

A floating point value that indicates the vertical position of the selected paired cursor.

Example

```
CURSor:PAIred:HPOS1?
```

This query might return `:CURSOR:PAIRED:HPOS1 -64.0000E-03`, indicating that the vertical position of cursor1 is at -64 mV.

CURSor:PAIred:POSITION<x>

Description

This command sets or queries the horizontal position (typically in time) of the cursor<x> paired cursor. This command is equivalent to selecting Cursor Position from the Cursors menu and then viewing or editing the desired cursor position.

Group

Cursor

Related Commands

CURSor:FUNCtion (see page 49)

Syntax 1

```
CURSor:PAIred:POSITION<x> <NRf>
```

Syntax 2

```
CURSor:PAIred:POSITION<x>?
```

Argument

- `<NRf>`

  This specifies the horizontal position of the cursor<x> paired cursor, which ranges from 1 to 2. The cursor position can appear in units of base or 1/base.

Example 1

```
CURSor:PAIred:POSITION1 9.0000E-06
```

This command specifies that the first paired cursor is at 9 µs.

Example 2

```
CURSor:PAIred:POSITION1?
```

This query might return `:CURSOR:PAIRED:POSITION 1.0000E-06`, indicating that the first paired cursor is at 1 µs.

CURSor:PAIred:UNIts?

Description

This query-only command requests the units for the paired cursors.

Group

Cursor

Syntax

```
CURSor:PAIred:UNIts?
```

Returns

The unit of measurement.

Example

```
CURSor:PAIred:UNIts?
```

This query might return: `:CURSOR:PAIRED:UNITS BASE`, indicating that base is the unit of measure.

CURSor:PAIred:VDELTA?

Description

This query-only command requests the Vbar (time) distance between paired cursors. It returns the absolute value of the first cursor less the second cursor horizontal positions. The position can appear in units of base and 1/base.

Group

Cursor

Syntax

```
CURSor:PAIred:VDELTA?
```

Returns

<NR3>

Example

```
CURSor:PAIred:VDELTA?
```

This query might return `:CURSOR:PAIRED:VDELTA 1.0640E+00`, indicating that time between the paired cursors is 1.064 s.

CURSor:SOUrce

Description

This command sets and queries the source for horizontal bar, vertical bar and paired cursors, and both sets and queries the source for split cursor1. This command is equivalent to selecting Cursor Setup from the Cursor menu and then choosing the desired cursor source.

Group

Cursor

Related Commands

CURSor:SPLit:SOURCE2 (see page 66)

Syntax 1

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

Syntax 2

```
CURSor:SOUrce?
```

Arguments

- `CH<x>`

  This is an input channel waveform. The valid channel waveform range is from 1 through 4.

- `MATH<x>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This is a reference waveform. The valid reference waveform range is from 1 through 4.

Returns

Any valid waveform.

Example 1

```
CURSor:SOUrce CH2
```

This command sets the cursor source to channel2.

Example 2

```
CURSor:SOUrce?
```

This query might return `:CURSOR:SOURCE CH2`, indicating that the cursor source is channel2.

CURSor:SPLit

Description

This command positions the Split cursors to positions defined by DATa:STARt and DATa:STOP (or maximum record length, if DATa:STOP is greater than the maximum record length). The query form of this command returns the units, positions and second source of the split cursors.

Group

Cursor

Related Commands

DATa:STARt (see page 473), DATa:STOP (see page 474)

Syntax 1

```
CURSor:SPLit SNAP
```

Syntax 2

```
CURSor:SPLit?
```

Argument

- `SNAP`

  This specifies that the vertical bar cursor positions will snap to DATa:STARt and DATa:STOP.

Example 1

```
CURSor:SPLit SNAP
```

This command specifies that the cursor positions will snap to DATa:STARt and DATa:STOP positions.

Example 2

```
CURSor:SPLit?
```

This query might return: `:CURSOR:SPLIT:POSITION1 -1.9992E+01 :POSITION2 -1.7996E+01;SOURCE2 CH1;UNITS BASE`.

CURSor:SPLit:HDELTA?

Description

This query-only command returns the vertical (volts) difference between cursor1 and cursor2. These values are represented on screen by the "X" markers (placed where the cursor marker and the waveform intersect).

Group

Cursor

Syntax

```
CURSor:SPLit:HDELTA?
```

Example

```
CURSor:SPLit:HDELTA?
```

This query might return `:CURSOR:SPLIT:HDELTA 3.1400E+00`, indicating that the vertical difference between two split cursors is 3.14 V.

CURSor:SPLit:HPOS<x>?

Description

This query-only command returns the vertical position of cursor<x> (i.e., that point where the Vbar intersects with the waveform), which can be specified as 1 or 2.

Group

Cursor

Related Commands

CURSor:SPLit:HDELTA (see page 63)

Syntax

```
CURSor:SPLit:HPOS<x>?
```

Example

```
CURSor:SPLit:HPOS1?
```

This query might return `:CURSOR:SPLIT:HPOS1 2.5400E+00`, indicating that the vertical position of cursor1 and cursor2 is 2.54 V.

CURSor:SPLit:POSITION<x>

Description

This command returns or sets the horizontal position of the cursor<x> vertical bar markers. This command is equivalent to selecting Cursor Setup from the Cursors menu and then entering the desired Cursor Position.

Group

Cursor

Syntax 1

```
CURSor:SPLit:POSITION<x>; <NRf>
```

Syntax 2

```
CURsor:SPLit:POSITION<x>?
```

Argument

- `<NRf>`

  This specifies the horizontal position of the cursor<x> cursor, which ranges from 1 to 2. The cursor position can appear in units of base or 1/base.

Example 1

```
CURSor:SPLit:POSITION2 350E-6
```

This command sets the position of cursor2 to 350 µs.

Example 2

```
CURSor:SPLit:POSITION1?
```

This query might return `:CURSOR:SPLIT:POSITION1 -204.0000E-09`, indicating the position of cursor1 is 204 ns.

CURSor:SPLit:SOURCE2

This command sets or returns the source waveform associated with split cursor2. This command is equivalent to selecting Cursor Setup from the Cursors menu and then entering the desired Cursor 2 Source.

Group

Cursor

Related Commands

CURSor:SOUrce (see page 61)

Syntax 1

```
CURSor:SPLit:SOURCE2
```

Syntax 2

```
CURsor:SPLit:SOURCE2?
```

Arguments

- `CH<x>`

  This is an input channel. The valid channel waveform range is from 1 through 4.

- `MATH<x>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This is a reference waveform. The valid reference waveform range is from 1 through 4.

Example 1

```
CURSor:SPLit:SOURCE2 CH3
```

This command sets cursor2 source to Channel3.

Example 2

```
CURSor:SPLit:SOURCE2?
```

This query might return `:CURSOR:SPLIT:SOURCE2 MATH2`, indicating that the source associated with cursor2 is Math2.

CURSor:SPLit:UNIts

Description

This command returns or sets the amplitude units of the split cursors.

Group

Cursor

Syntax 1

```
CURSor:SPLit:UNIts
```

Syntax 2

```
CURsor:SPLit:UNIts?
```

Example 1

```
CURSor:SPLit:UNIts BASE
```

This command sets the amplitude units of the split cursors to base.

Example 2

```
CURSor:SPLit:UNIts?
```

This query would return `:CURSOR:SPLIT:UNITS BASE`, indicating that base is the current setting for split cursor amplitude units.

CURSor:SPLit:VDELTA?

Description

This command returns the horizontal (time) difference between the split cursors. It returns the absolute value of the first cursor less the second cursor horizontal positions. Note that the split cursor values are for source1 and source2, and the sources can be located on different waveforms. The position can appear in units of base and 1/base.

Group

Cursor

Syntax

```
CURsor:SPLit:VDELTA?
```

Example

```
CURSor:SPLit:VDELTA?
```

This command might return `:CURSOR:SPLIT:VDELTA 1.0640E+00`, indicating that the time between the vertical bar cursors is 1.064 s.

CURSor:STATE

Description

This command sets or returns the state of the cursor. The cursor can either be turned on or off. 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.

Cursor

Related Commands

CURSor:FUNCtion OFF (see page 49)

Syntax 1

```
CURSor:STATE {ON|OFF}
```

Syntax 2

```
CURsor:STATE?
```

Arguments

- `ON`

  This activates the cursor on display.

- `OFF`

  This removes the cursor from display.

Example 1

```
CURSor:STATE ON
```

This command displays the cursor.

Example 2

```
CURSor:STATE?
```

This query might return `:CURSOR:STATE ON`, indicating that the cursor is active on the display.

CURSor:VBArs

Description

This command positions the vertical bar cursor to the position defined by DATa:STARt or DATa:STOP .
This query returns the current vertical bar settings for horizontal position and units.

Group

Cursor

Related Commands

DATa:STARt (see page 473), DATa:STOP (see page 474)

Syntax 1

```
CURSor:VBArs?
```

Syntax 2

```
CURSor:VBArs SNAp
```

Argument

- `SNAp`

  This positions the vertical bar cursors at DATa:STARt and DATa:STOP.

Example 1

```
CURSor:VBArs?
```

This query might return `:CURSOR:VBARS:UNITS SECONDS ;POSITION1 1.0000E-06;POSITION2 9.0000E-06`

Example 2

```
CURSor:VBArs SNAp
```

This command positions the vertical bar cursor to the position defined by DATa:STARt or DATa:STOP.

CURSor:VBArs:DELTa?

Description

This query-only command returns the 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

Related Commands

CURSor:VBArs:UNIts (see page 73)

Syntax

```
CURSor:VBArs:DELTa?
```

Returns

A floating point value with an exponent.

Example

```
CURSor:VBArs:DELTa?
```

This command 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>

Description

This command sets or queries the horizontal position for both vertical bar and paired cursors. 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 the V Bars Cursor Type and then viewing or editing the desired cursor position.

Group

Cursor

Related Commands

CURSor:VBArs:UNIts (see page 73)

Syntax 1

```
CURSor:VBArs:POSITION<x> <NRf>
```

Syntax 2

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

Argument

<NRf>

Specifies the cursor position.

Returns

A floating point value with an exponent.

Example 1

```
CURSor:VBArs:POSITION2 9.00E-6
```

Positions the cursor2 vertical bar cursors at 9 ms.

Example 2

```
CURSor:VBArs:POSITION1?
```

This command might return `:CURSOR:VBARS:POSITION1 1.0000E-06`, indicating the cursor1 vertical bar is at 1 ms.

CURSor:VBArs:UNIts

Description

This command sets or queries the units for the vertical bar cursors.

Group

Cursor

Syntax 1

```
CURSor:VBArs:UNIts
```

Syntax 2

```
CURSor:VBArs:UNIts?
```

Arguments

- `SECOnds`

  Fix the units for the VBArs for time domain.

- `HERtz`

  Fix the units for the VBArs for frequency domain.

Returns

A string for SECOnds or HERtz, depending upon current vertical bar cursor units.

Example 1

```
CURSor:VBArs:UNIts HERtz
```

Sets the units for the VBArs cursors to 1/seconds.

Example 2

```
CURSor:VBArs:UNIts?
```

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

Diagnostics Command Group

Diagnostics Overview

You use the commands in the Diagnostic Command Group to control the selection and execution of diagnostic tests. You can use Test functions to select and execute an item at any level of the test hierarchy. Control functions allow controls for diagnostic execution to be set. RSI-like functions provide the same capability of selecting and running diagnostic tests that are normally available from the RSI.

Diagnostics Commands

Command	Description
DIAg:CONTROl:HALT	Enables or disables halting on first diagnostic failure
DIAg:CONTROl:LOOp	Enables or disables looping of diagnostics
DIAg:EXECUTE	Executes currently selected set of diagnostics
DIAg:ITEM?	Returns all data associated with a particular item of the current RSI menu
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 result from the last execution of the 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 that 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:PUMODE?	Returns the current diagnostic mode of the instrument
DIAg:PUMODE	Sets the current diagnostic mode of the instrument
DIAg:RESUlts?	Returns a brief pass or fail status of the last diagnostic execution
DIAG:RESUlts:VERBose?	Returns a more explanatory message about the results of the last diagnostic execution
DIAg:SELect:ALL	Selects all available diagnostics
DIAg:SELect:AREA	Selects one of the available diagnostic areas
DIAg:SELect:LAST	Sets the last item of a group of items

	from the same level of test hierarchy
DIAg:SELect:SUBSYS	Selects one of the available diagnostic subsystems
DIAg:SELect:TEST	Selects one of the available diagnostic tests
DIAg:STATE	Sets the instrument operating state
DIAg:STOP	Terminates the execution of diagnostics
ERRLOG CLEAR	Deletes all entries in the error log and zeroes the test sequence count
ERRLOG:FIRST?	Returns the first message in the error log
ERRLOG:NEXT?	Returns the next message in the error log
ERRLOG:NUMENTRIES?	Returns the total number of messages in the error log
TEST	Selects and executes any 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

DIAg:CONTROL:HALT

Description

This command (no query form) determines 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

Related Commands

DIAg:CONTROL:LOOP (see page 78)

Syntax

```
DIAg:CONTROL:HALT (ON|OFF|<NR1>}
```

Arguments

- `ON`

  This enables the halt function, causing the execution of diagnostics to halt at the first diagnostic failure that occurs.

- `OFF`

  This disables the halt function, allowing the oscilloscope to execute the entire set of diagnostics before halting, even if diagnostic failure occurs.

- `<NR1>`

  A 0 enables the halt function; any other value disables the halt function.

Example

```
DIAg:CONTROL:HALT ON
```

This command enables the halt function, causing the execution of diagnostics to halt at the first diagnostic failure that occurs.

DIAg:CONTROL:LOOP

Description

This command (no query form) determines 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

Related Commands

DIAg:CONTROL:HALT (see page 77), DIAg:STOP (see page 101)

Syntax

```
DIAg:CONTROL:LOOP {ON|OFF|<NR1>}
```

Arguments

- `ON`

  This enables the loop function, causing the execution of diagnostics to continuously loop.

- `OFF`

  This disables the loop function, causing the oscilloscope to execute the entire set of diagnostics once and then halt.

- `<NR1>`

  A 0 enables the loop function; any other value disables the loop function.

Example

```
DIAg:CONTROL:LOOP ON
```

This command enables the loop function.

DIAg:EXECUTE

Description

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

Related Commands

DIAg:STATE (see page 100)

Syntax

```
DIAg:EXECUTE
```

Example

```
DIAg:EXECUTE
```

This command starts the execution of the entire set of diagnostics.

DIAg:ITEM?

Description

This query-only command returns all the data associated with a particular item from the RSI menu (level of test hierarchy). This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the diagnostics settings.

Group

Diagnostics

Related Commands

DIAg:ITEM:FAILURES? (see page 81), DIAg:ITEM:NAME? (see page 82), DIAg:ITEM:RESULT? (see page 83), DIAg:ITEM:SUBITEMS? (see page 84)

Syntax

```
DIAg:ITEM?
```

Arguments

- `<NR1>`

  This sets the index item about which data will be returned, which ranges from 0 through 14.

Example

```
DIAg:ITEM 2?
```

This query might return `:DIAG:ITEM "TRIGGER","FAIL",2,7`

DIAg:ITEM:FAILURES?

Description

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.

Group

Diagnostics

Related Commands

DIAg:ITEM? (see page 80), DIAg:ITEM:NAME? (see page 82), DIAg:ITEM:RESULT? (see page 83), DIAg:ITEM:SUBITEMS? (see page 84)

Syntax

```
DIAg:ITEM:FAILURES?
```

Example

```
DIAg:ITEM:FAILURES?
```

This query might return `:DIAG:ITEM:FAILURES 2`, indicating the number of failures.

DIAg:ITEM:NAMe?

Description

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.

Group

Diagnostics

Related Commands

DIAg:ITEM? (see page 80), DIAg:ITEM:FAILURES? (see page 81), DIAg:ITEM:RESULT? (see page 83), DIAg:ITEM:SUBITEMS? (see page 84)

Syntax

```
DIAg:ITEM:NAMe?
```

Example

```
DIAg:ITEM:NAMe?
```

This query might return `:DIAG:ITEM:NAME "TRIGGER"`, indicating that the selected menu item is Trigger.

DIAg:ITEM:RESULT?

Description

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.

Group

Diagnostics

Related Commands

DIAg:ITEM? (see page 80), DIAg:ITEM:NAMe? (see page 82), DIAg:ITEM:FAILURES? (see page 81), DIAg:ITEM:SUBITEMS? (see page 84)

Syntax

```
DIAg:ITEM:RESULT?
```

Example 1

```
DIAg:ITEM:RESULT?
```

This query might return `:DIAG:ITEM:RESULT "PASS"`, indicating that the item passed during the last execution.

Example 2

```
DIAg:ITEM:RESULT?
```

This query might return `:DIAG:ITEM:RESULT "FAIL"`, indicating that the item failed during the last execution.

Example 3

```
DIAg:ITEM:RESULT?
```

This query might return `:DIAG:ITEM:RESULT "*****"`, indicating that the item was not run.

DIAg:ITEM:SUBITEMS?

Description

This query-only command returns the number of sub-items 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 sub-items.

Group

Diagnostics

Related Commands

DIAg:ITEM? (see page 80), DIAg:ITEM:NAME? (see page 82), DIAg:ITEM:FAILURES? (see page 81), DIAg:ITEM:RESULT? (see page 83)

Syntax

```
DIAg:ITEM:SUBITEMS?
```

Example

```
DIAg:ITEM:SUBITEMS?
```

This query might return `:DIAG:ITEM:SUBITEMS 15`, indicating that there are 15 sub-items associated with the selected item.

DIAg:LEVEL

Description

This command sets or returns the currently 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 1

```
DIAg:LEVEL (SUBSYS|AREA|TEST}
```

Syntax 2

```
DIAg:LEVEL?
```

Arguments

- `SUBSYS`

  This sets diagnostic testing to the subsystem level.

- `AREA`

  This sets diagnostic testing to the area level.

- `TEST`

  This sets diagnostic testing to the test level.

Example 1

```
DIAg:LEVEL AREA
```

This command sets the level of diagnostic test hierarchy to Area.

Example 2

```
DIAg:LEVEL?
```

This query might return `:DIAG:LEVEL:SUBSYS`, indicating that the current level of diagnostic test hierarchy is Subsys.

DIAg:LOOPS?

Description

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 Loop Count.

Group

Diagnostics

Syntax

```
DIAg:LOOPS?
```

Example

```
DIAg:LOOPS?
```

This query might return `:DIAG:LOOPS 5`, indicating that the selected set of diagnostics was completed five times during the last diagnostic execution.

DIAg:NAMe?

Description

This query-only command returns the names of the subsystem, area, and test of the current diagnostic test.

Group

Diagnostics

Related Commands

DIAg:NAMe:AREA? (see page 88), DIAg:NAMe:SUBSYS? (see page 89), DIAg:NAMe:TEST? (see page 90)

Syntax

```
DIAg:NAMe?
```

Example

```
DIAg:NAMe?
```

This query might return `:DIAG:NAME:SUBSYS "Acquisition";AREA "Memory";TEST"diag Data Format"`, indicating the subsystem name, area name, and test name of the currently selected diagnostic test.

DIAg:NAMe:AREA?

Description

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.

Group

Diagnostics

Related Commands

DIAg:NAMe? (see page 82), DIAg:NAMe:SUBSYS? (see page 89), DIAg:NAMe:TEST? (see page 90)

Syntax

```
DIAg:NAMe:AREA?
```

Example

```
DIAg:NAMe:AREA?
```

This query might return `:DIAG:NAME:AREA "Memory"`, indicating the selected area name of the current diagnostic test.

DIAg:NAMe:SUBSYS?

Description

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.

Group

Diagnostics

Related Commands

DIAg:NAMe? (see page 87), DIAg:NAMe:AREA? (see page 88), DIAg:NAMe:TEST? (see page 90)

Syntax

```
DIAg:NAMe:SUBSYS?
```

Example

```
DIAg:NAMe:SUBSYS?
```

This query might return `:DIAG:NAME:SUBSYS "Acquisition"`, indicating the subsystem name of the current diagnostic test.

DIAg:NAMe:TEST?

Description

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.

Group

Diagnostics

Related Commands

DIAg:NAMe? (see page 87), DIAg:NAMe:AREA? (see page 88), DIAg:NAMe:SUBSYS? (see page 89)

Syntax

```
DIAg:NAMe:TEST?
```

Example

```
DIAg:NAMe:TEST?
```

This query might return `:DIAG:NAME:TEST "diagDataFormatConf"`, indicating the test name of the current diagnostic test.

DIAg:NUMITEMS?

Description

This query-only command returns the number of items on the currently selected level of test hierarchy, which ranges from 0 through 15. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

Group

Diagnostics

Syntax

```
DIAg:NUMITEMS?
```

Example

```
DIAg:NUMITEMS?
```

This query might return `:DIAG:NUMITEMS 7`, indicating the number of items on the currently selected level of test hierarchy.

DIAg:PUMODE

Description

This command sets or returns the current diagnostic mode of the instrument.

Group

Diagnostics

Syntax 1

```
DIAg:PUMODE {USER|FACTORY}
```

Syntax 2

```
DIAg:PUMODE?
```

Arguments

- `USER`

  This sets the instrument to user diagnostic mode.

- `FACTORY`

  This sets the instrument to factory diagnostic mode.

Example 1

```
DIAg:PUMODE FACTORY
```

This command sets the instrument to the factory diagnostic mode.

Example 2

```
DIAg:PUMODE?
```

This query might return `:DIAG:PUMODE USER`, indicating that instrument is currently set to user diagnostic mode.

DIAg:RESULts?

Description

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.

Group

Diagnostics

Related Commands

DIAg:RESULts:VERBose? (see page 94)

Syntax

```
DIAg:RESUlts?
```

Example 1

```
DIAg:RESULts?
```

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

Example 2

```
DIAg:RESULts?
```

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

Example 3

```
DIAg:RESULts?
```

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

DIAg:RESULts:VERBose?

Description

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.

Group

Diagnostics

Related Commands

DIAg:RESUlts? (see page 93)

Syntax

```
DIAg:RESUlts:VERBose?
```

Example

```
DIAg:RESUlts:VERBose?
```

This query might return `:DIAG:RESULTS:VERBOSE` `"DIAG ABORTED, ERROR 15 in A2D-Dmux 1 interconnects"`

DIAg:SELect:ALL

Description

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

Related Commands

DIAg:NAMe:AREA? (see page 88), DIAg:NAMe:SUBSYS? (see page 89), DIAg:NAMe:TEST? (see page 90), DIAg:EXECUTE (see page 79)

Syntax

```
DIAg:SELect:ALL
```

Example

```
DIAg:SELect:ALL
```

This command selects all available diagnostics.

DIAg:SELect:AREA

Description

This command (no query form) selects one of the available areas. 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

Related Commands

DIAg:SELect:SUBSYS (see page 98), DIAg:SELect:TEST (see page 99), DIAg:SELect:LAST (see page 97), DIAg:NUMITEMS (see page 91)

Syntax

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

Argument

- `<NR1>`

  This selects an area by number, which can range from 1 through 15 (as specified by DIAg:NUMITEMS).

Example

```
DIAg:SELect:AREA 1
```

This command selects Area 1 as the area to be diagnosed.

DIAg:SELect:LAST

Description

This command (no query form) 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, specifying `DIAg:SELECT:SUBSYS 2` and `DIAg:NUMITEMS 5` 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

Related Commands

DIAg:EXECUTE (see page 79), DIAg:LEVEL (see page 85), DIAg:NAMe:AREA? (see page 88), DIAg:NAMe:SUBSYS? (see page 89), DIAg:NAMe:TEST? (see page 90), DIAg:NUMITEMS? (see page 91), DIAg:SELect:SUBSYS (see page 98)

Syntax

```
DIAg:SELect:LAST <NR1>
```

Arguments

- `<NR1>`

  This selects an integer that identifies the number of the last item that will be executed when the DIAg:EXECUTE command is run.

Example

```
DIAg:SELect:LAST 2
```

This command specifies that (based on the previous example) only subsystem 2 will be executed.

DIAg:SELect:SUBSYS

Description

This command (no query form) selects one of the available subsystems. 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

Related Commands

DIAg:SELect:AREA (see page 96), DIAg:SELect:TEST (see page 99), DIAg:SELect:LAST (see page 97), DIAg:NUMITEMS (see page 91)

Syntax

```
DIAg:SELect:SUBSYS <NR1>
```

Argument

- `<NR1>`

  This selects a subsystem by number, which can range from 1 through 15 (as specified by DIAg:NUMITEMS).

Example

```
DIAg:SELect:SUBSYS
```

This command selects Subsystem 1 as the subsystem to be diagnosed.

DIAg:SELect:TEST

Description

This command (no query form) selects 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

Related Commands

DIAg:SELect:AREA (see page 96), DIAg:SELect:SUBSYS (see page 98), DIAg:SELect:LAST (see page 97), DIAg:NUMITEMS (see page 91)

Syntax

```
DIAg:SELect:TEST <NR1>
```

Arguments

- `<NR1>`

  This selects a test by number, which can range from 1 through 15 (as specified by DIAg:NUMITEMS).

Example

```
DIAg:SELect:TEST 1
```

This command selects Test 1 as the test to be executed.

DIAg:STATE

Description

This command (no query form) changes the oscilloscope 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

Related Commands

TEST:STOP (see page 105)

Syntax

```
DIAg:STATE {ON|OFF}
```

Arguments

- `ON`

  This puts the instrument into the state in which diagnostics can be run. This argument is thrown automatically if either the DIAg:EXECUTE or DIAg:TEST commands are executed.

- `OFF`

  This disables diagnostics capabilities and returns the oscilloscope to a normal operating state.

Example

```
DIAg:STATE OFF
```

This command turns off diagnostics capabilities and returns the instrument to a normal operating state.

DIAg:STOP

Description

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

Related Commands

TEST:STOP (see page 105)

Syntax

```
DIAg:STOP
```

Example

```
DIAg:STOP
```

This command terminates diagnostics (or test) execution at the end of the next low-level test.

TEST

Description

This command (no query form) provides the ability to select and execute an item at any level of the test hierarchy (Test, Area or Subsystem). 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

Related Commands

DIAg:CONTROL:HALT (see page 77), DIAg:CONTROL:LOOP (see page 78), TEST:RESults? (see page 103), TEST:RESults:VERBose? (see page 104), TEST:STOP (see page 105)

Syntax

```
TEST
```

Argument

- `<NR3>`

  This sets the test ID, which ranges from 0 through 3 characters. If no test ID is specified, all available diagnostics are executed.

Example 1

```
TEST "1"
```

This command executes all Acquisition tests (Subsystem:Acquisition).

Example 2

```
TEST "11"
```

This command executes all Acquisition Memory tests (Subsystem:Acquisition, Area:Memory)

Example 3

```
TEST "113"
```

This command executes a specific Acquisition Memory test (Subsystem:Acquisition, Area:Memory, Test:Address Lines).

TEST:RESults?

Description

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

Related Commands

TEST (see page 102), TEST:RESults:VERBose? (see page 104)

Syntax

```
TEST:RESults?
```

Example 1

```
TEST:RESults?
```

This query might return `:TEST:RESULTS "135"`, indicating the specific failure code of the first test failure.

Example 2

```
TEST:RESults?
```

This query might return `:TEST:RESULTS "PASS"`, indicating that the previously executed test passed.

Example 3

```
TEST:RESults?
```

This query might return `:TEST:RESULTS "*****"`, indicating that the test was not run.

TEST:RESults:VERBose?

Description

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

Related Commands

TEST (see page 102), TEST:RESults? (see page 103)

Syntax

```
TEST:RESults:VERBose?
```

Example

```
TEST:RESults:VERBose?
```

This query might return `:TEST:RESULTS:VERBOSE DIAG ABORTED, ERROR 15 in A2D-Dmux 1 interconnects"`

TEST:STOP

Description

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

Related Commands

DIAg:STOP (see page 101)

Syntax

```
TEST:STOP
```

Example

```
TEST:STOP
```

This command terminates test (or diagnostics) execution at the end of the next low-level test.

ERRLOG CLEAR

Description

This command (no query form) deletes all entries in the log and zeroes the test sequence count. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, clicking Error Log and then clicking Clear Log.

Group

Diagnostics

Related Commands

ERRLOG:FIRST? (see page 107), ERRLOG:NEXT? (see page 108), ERRLOG:NUMENTRIES? (see page 109)

Syntax

```
ERRLOG CLEAR
```

Arguments

- `CLEAR`

  This removes all entries in the error log and zeroes the test sequence count.

Example

```
ERRLOG CLEAR
```

This command deletes all entries in the log and zeroes the test sequence count.

ERRLOG:FIRST?

Description

This command query returns the first message in the error log. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking the Error Log button.

Group

Diagnostics

Related Commands

ERRLOG CLEAR (see page 106), ERRLOG:NEXT? (see page 108), ERRLOG:NUMENTRIES? (see page 109)

Syntax

```
ERRLOG:FIRST?
```

Example

```
ERRLOG:FIRST?
```

This command might return `:ERRLOG:FIRST "01JAN70 00:00:41 P Pu=1 T=26 Test id=323 digSpeedMemConf: ""U406 addr(0x4cff5) = 0x04e8b5 act/exp = 0x7e00/0x5e00"""`, which is the first message in the error log.

ERRLOG:NEXT?

Description

This command query returns the next message in the error log. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking the Error Log button.

Group

Diagnostics

Related Commands

ERRLOG CLEAR (see page 106), ERRLOG:FIRST? (see page 107), ERRLOG:NUMENTRIES? (see page 109)

Syntax

```
ERRLOG:NEXT?
```

Example

```
ERRLOG:NEXT?
```

This command might return `:ERRLOG:NEXT "01JAN70 00:00:41 P Pu=1 T=26 Test id=323 digSpeedMemConf: ""U406 addr(0x4cff5) = 0x04e8b5 act/exp = 0x7e00/0x5e00"""`, which is the next message in the error log.

ERRLOG:NUMENTRIES?

Description

This command query returns the total number of messages in the error log. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then viewing the Error Log.

Group

Diagnostics

Related Commands

ERRLOG CLEAR (see page 106), ERRLOG:FIRST? (see page 107), ERRLOG:NEXT? (see page 108)

Syntax

```
ERRLOG:NUMENTRIES?
```

Example

```
ERRLOG:NUMENTRIES?
```

This query might return `:ERRLOG:NUMENTRIES 3`, indicating the total number of entries in the error log.

Display Control Command Group

Display Overview

You 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 following:

- Background and foreground color as well as cursor, histogram, mask, and measurement annotation colors.
- Display of date and time; cursor, histogram, mask, and measurement readouts; measurement annotations, and the mode in which waveforms are displayed.
- Interpolation to increase sample density of waveform for high acquisition rates

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.

Display Control Commands

Command	Description
DISplay?	Returns current display settings
DISplay:CLOCk?	Sets the display of the date/time stamp
DISplay:CLOCk	Returns the display of the date/time stamp
DISplay:COLOr?	Returns color group settings
DISplay:COLOr:MATHCOLOR?	Returns the color to be used for math traces
DISplay:COLOr:MATHCOLOR	Sets the color to be used for math traces
DISplay:COLOr:PALETTE?	Returns the palette to be used for trace display
DISplay:COLOr:PALETTE	Sets the palette to be used for trace display
DISplay:COLOr:REFCOLOR?	Returns the color to be used for reference traces
DISplay:COLOr:REFCOLOR	Sets the color to be used for reference traces
DISplay:FILTer?	Returns the type of interpolation to use for the display
DISplay:FILTer	Sets the type of interpolation to use for the display
DISplay:FORMat?	Returns the display format
DISplay:FORMat	Sets the display format
DISplay:GRAticule?	Returns the type of graticule that is displayed
DISplay:GRAticule	Sets the type of graticule that is displayed
DISplay:INTENSITy?	Returns the waveform saturation level, autobright state and screensaver settings
DISplay:INTENSITy:AUTOBRIGHT?	Returns the autobright state
DISplay:INTENSITy:AUTOBRIGHT	Sets auto-leveling of waveform saturation
DISplay:INTENSITy:SCREENSAVER?	Returns the display protection feature state
DISplay:INTENSITy:SCREENSAVER	Enables automatic display protection features
DISplay:INTENSITy:SCREENSAVER DELAY?	Returns the display protection timeout state
DISplay:INTENSITy:SCREENSAVER DELAY	Sets the display protection timeout
DISplay:INTENSITy:WAVEform?	Returns the intensity saturation of the waveforms
DISplay:INTENSITy:WAVEform	Sets the intensity saturation of the waveforms

DISplay:PERSistence?	Returns display persistence setting
DISplay:PERSistence	Sets display persistence setting
DISplay:STYLE?	Returns data display style
DISplay:STYLE	Sets data display style
DISplay:TRIGBar?	Returns the display setting of the trigger level indicator bar(s)
DISplay:TRIGBar	Sets the display of the trigger level indicator bar(s)
DISplay:TRIGT?	Returns the display setting of the trigger point indicator
DISplay:TRIGT	Sets the display of the trigger point indicator
DISplay:VARPersist?	Returns the persistence decay time
DISplay:VARPersist	Sets the persistence decay time

DISplay?

Description

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?
```

Example

```
DISplay?
```

This query might return

```
:DISPLAY:CLOCK 1
;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT
;REFCOLOR DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT
;GRATICULE FULL;INTENSITY:WAVEFORM 6.0000E+01
;AUTOBRIGHT 1;SCREENSAVER 1;SCREENSAVERDELAY 28800,
:DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT
;TRIGT 0;VARPERSIST 5.0000E-01
```

DISplay:CLOCk

Description

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 1

```
DISplay:CLOCk {ON|OFF|<NR1>}
```

Syntax 2

```
DISplay:CLOCk?
```

Arguments

- `ON`

  This enables display of date and time.

- `OFF`

  This disables display of date and time.

- `<NR1>`

  A 0 disables display of date and time; any other value enables display of date and time.

Example 1

```
DISplay:CLOCk ON
```

This command enables display of date and time.

Example 2

```
DISplay:CLOCk?
```

This query might return `:DISPLAY:CLOCK 1`, indicating that the display of date and time is currently enabled.

DISplay:COLOr?

Description

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.

Example

```
DISplay:COLOr?
```

This query might return `:DISPLAY:COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR INHERIT`

DISplay:COLOr:PALETTE

Description

This command sets or queries the palette to be used for trace display. This is equivalent to selecting Colors from the Display menu and choosing from the displayed color palette options.

Group

Display Control

Related Commands

DISplay:COLOr (see page 115), DISplay:COLOr:MATHCOLOr (see page 117), DISplay:COLOr:REFCOLOr (see page 118)

Syntax 1

```
DISplay:COLOr:PALETTE {NORMal|MONOGREEN|MONOGRAY|TEMPErature|SPECTral}
```

Syntax 2

```
DISplay:COLOr:PALETTE?
```

Arguments

- `NORMal`

  This colors traces according to their channel. This is the default color palette.

- `MONOGREEN`

  This colors traces green, emulating a traditional oscilloscope color palette.

- `MONOGRAY`

  This colors traces gray, emulating a monochrome TDS instrument.

- `TEMPErature`

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

- `SPECTral`

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

Example 1

```
DISplay:COLOr:PALETTE MONOGRAY
```

This command sets the palette for all display types.

Example 2

```
DISplay:COLOr:PALETTE?
```

This query might return `:DISPLAY:COLOR:PALETTE NORMAL`, indicating that the Normal color palette is currently selected, which colors traces according to their channel.

DISplay:COLOr:MATHCOLOr

Description

This command sets or queries the color to be used for math traces, either in the 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

Related Commands

DISplay:COLOr:PALETTE (see page 116)

Syntax 1

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

Syntax 2

```
DISplay:COLOr:MATHCOLOr?
```

Arguments

- `DEFAULT`

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

- `INHERIT`

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

Example 1

```
DISplay:COLOr:MATHCOLOr DEFAULT
```

This command sets the color math traces in the nominal palette math color.

Example 2

```
DISplay:COLOr:MATHCOLOr?
```

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

DISplay:COLOr:REFCOLOr

Description

This command sets or queries the color to be used for reference traces, either in the 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

Related Commands

DISplay:COLOr:PALETTE (see page 116)

Syntax 1

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

Syntax 2

```
DISplay:COLOr:REFCOLOr?
```

Arguments

- `DEFAULT`

  This assigns color reference traces in nominal palette reference color, which is off-white.

- `INHERIT`

  This assigns color reference traces in the source waveform color.

Example 1

```
DISplay:COLOr:REFCOLOr DEFAULT
```

This command assigns color reference traces in nominal palette reference color, which is off-white.

Example 2

```
DISplay:COLOr:REFCOLOr?
```

This query might return `:DISPLAY:COLOR:REFCOLOR INHERIT`, indicating that source waveform color is used for reference traces.

DISplay:FILTer

Description

This command sets or queries the type of interpolation to use for the display. Filtering only applies to normal-mode acquisition. DPO never interpolates. 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 1

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

Syntax 2

```
DISplay:FILTer?
```

Arguments

- `LINEAr`

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

- `SINX`

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

Example 1

```
DISplay:FILTer SINX
```

This command specifies sine-curve interpolation when magnifying waveforms.

Example 2

```
DISplay:FILTer?
```

This query might return `:DISPLAY:FILTER LINEAR`, indicating that straight-line interpolation is specified for magnifying waveforms.

DISplay:FORMat

Description

This command sets or queries the display format. This command is equivalent to setting Format from the Display menu.

Group

Display Control

Syntax 1

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

Syntax 2

```
DISplay:FORMat?
```

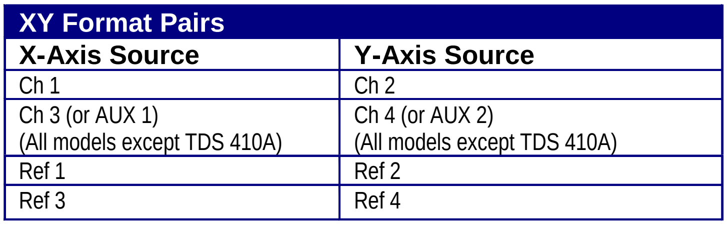
Arguments

- `YT`

  This sets the display to a voltage versus time format and is the normal mode.

- `XY`

  This 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.



XY Format Pairs

X-Axis Source	Y-Axis Source
Ch 1	Ch 2
Ch 3 (or AUX 1) (All models except TDS 410A)	Ch 4 (or AUX 2) (All models except TDS 410A)
Ref 1	Ref 2
Ref 3	Ref 4

  Note: Selecting XY or XYZ forces a switch to FASTAcq mode.

- `XYZ`

  This combines channel 1 and channel 2 for X and Y coordinates and uses channel 3 to provide the intensity value for the sample. XYZ groups channels 1, 2 and 3 to form a single trace. Other channel, math and reference waveforms are turned off.

Example 1

```
DISplay:FORMat YT
```

This command selects a voltage versus time format for the display.

Example 2

```
DISplay:FORMat?
```

This query might return `:DISPLAY:FORMAT XYZ` for the display format.

DISplay:GRATicule

Description

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 1

```
DISplay:GRAticule {CROSSHair|FRAme|FULl|GRId}
```

Syntax 2

```
DISplay:GRAticule?
```

Arguments

- `CROSSHair`

  This specifies a frame and cross hairs.

- `FRAme`

  This specifies just a frame.

- `FULl`

  This specifies a frame, a grid and cross hairs.

- `GRId`

  This specifies a frame and grid only.

Example 1

```
DISplay:GRAticule FRAme
```

This command sets the graticule type to display the frame only.

Example 2

```
DISplay:GRAticule?
```

This query might return `:DISPLAY:GRATICULE FULL`, indicating that all graticule elements are selected.

DISplay:INTENSITy?

Description

This query-only command returns the waveform saturation level, autobright state and screensaver settings. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Appearance tab.

Group

Display Control

Related Commands

DISplay:INTENSITy:AUTOBright (see page 123), DISplay:INTENSITy:WAVEform (see page 124), DISplay:INTENSITy:SCREENSAVER (see page 125), DISplay:INTENSITy:SCREENSAVERDELAY (see page 126)

Syntax

```
DISplay:INTENSITy?
```

Arguments

None

Example

```
DISplay:INTENSITy?
```

This query might return `:DISPLAY:INTENSITY:WAVEFORM 7.5000E+01; AUTOBRIGHT 1; SCREENSAVER 1; SCREENSAVERDELAY 120`

DISplay:INTENSITy:AUTOBright

Description

This command enables automatic, ongoing adjustment of the hi-res to 4-bit-display mapping transform for converting DPO or RecordVu images to display images. The query form returns an ON (1) or OFF (0). This command is equivalent to selecting Display Setup from the Display menu and then choosing the Appearance tab.

Group

Display Control

Related Commands

DISplay:INTENSITy:WAVEform (see page 124)

Syntax 1

```
DISplay:INTENSITy:AUTOBright {OFF|ON|<NR1>}
```

Syntax 2

```
DISplay:INTENSITy:AUTOBright?
```

Arguments

- `OFF`

  This allows the system to use the manually set waveform intensity value against an absolute scale.

- `ON`

  This allows the system to adjust settings with the user-defined waveform intensity value in concert with the maximum and minimum non-zero acquired pixel intensity.

- `<NR1>`

  A 0 allows the system to use the manually set waveform intensity value against an absolute scale; any other value allows the system to adjust settings.

Example 1

```
DISplay:INTENSITy:AUTOBright ON
```

This command allows the system to adjust settings using the waveform intensity value and the pixel intensity found in the acquired waveform.

Example 2

This query might return `:DISPLAY:INTENSITY:AUTOBRIGHT 1`, indicating that system adjustment of waveform intensity is enabled.

DISplay:INTENSITy:WAVEform

Description

This command sets or queries the intensity saturation of the waveforms. This command is equivalent adjusting the **INTENSITY** knob on the front panel.

Group

Display Control

Related Commands

DISplay:INTENSITy:AUTOBright (see page 123)

Syntax 1

```
DISplay:INTENSITy:WAVEform <NR1>
```

Syntax 2

```
DISplay:INTENSITy:WAVEform?
```

Arguments

- `<NR1>`

  This is the waveform intensity and ranges from 10 to 100 percent.

Example 1

```
DISplay:INTENSITy:WAVEform 100
```

This command sets the waveform intensity to 100 percent.

Example 2

```
DISplay:INTENSITy:WAVEform?
```

This query might return `:DISPLAY:INTENSITY:WAVEFORM 6.0000E+01`, indicating that the intensity of the waveforms is currently set to 60 percent.

DISplay:INTENSITy:SCREENSAVER

Description

This command sets and queries the screen protection features of the underlying MS Windows operating system. When enabled, a delay timer (set in seconds by the DISplay:INTENSITy:SCREENSAVERDELAY command) begins counting down. When this screensaver delay times out, the screen low-power mode engages. This causes the LCD backlight to switch off and clears both waveform and text displays. Any control (front panel, mouse or keyboard) or touch screen activity resets the delay timer and restores normal scope display. This command is equivalent to selecting Screen Saver Enabled from the Display menu.

Group

Display Control

Related Commands

DISplay:INTENSITy:SCREENSAVERDELAY (see page 126)

Syntax 1

```
DISplay:INTENSITy:SCREENSAVER {OFF|ON|<NR1>}
```

Syntax 2

```
DISplay:INTENSITy:SCREENSAVER?
```

Arguments

- `OFF`

  This disables the screen saver feature.

- `ON`

  This enables the protection after the specified screen saver delay seconds of control activity have passed.

- `<NR1>`

  A 0 disables the screensaver feature; a 1 enables the screen saver protection features.

Example 1

```
DISplay:INTENSITy:SCREENSAVER OFF
```

This command disables the screen saver feature.

Example 2

```
DISplay:INTENSITy:SCREENSAVER?
```

This query might return `:DISPLAY:INTENSITy:SCREENSAVER 0`, indicating that the screensaver feature is disabled.

DISplay:INTENSITy:SCREENSAVERDELAY

Description

This command sets or queries the timeout of the screen protection features of the display system. When enabled (after the specified screen saver delay seconds of control activity and when the screensaver feature is enabled) the instrument activates the screensaver feature. Normal scope displays are restored and the delay timer is reset upon any control activity. The oscilloscope continues to acquire and process data normally while in screensaver mode; only the display is disabled. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Objects tab.

Group

Display Control

Related Commands

DISplay:INTENSITy:SCREENSAVER (see page 125)

Syntax 1

```
DISplay:INTENSITy:SCREENSAVERDELAY {<NR1>}
```

Syntax 2

```
DISplay:INTENSITy:SCREENSAVERDELAY?
```

Arguments

- `<NR1>`

  This sets the screensaver timeout, which ranges from 30 through 28800 seconds.

Example 1

```
DISplay:INTENSITy:SCREENSAVERDELAY 120
```

This command sets the screen saver delay feature to activate after 120 seconds of control inactivity.

Example 2

```
DISplay:INTENSITy:SCREENSAVERDELAY?
```

This query might return `:DISPLAY:INTENSITy:SCREENSAVERDELAY 300`, indicating that the screen saver delay feature is set to 300 seconds of control inactivity.

DISplay:PERSistence

Description

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

Related Commands

DISplay:VARPersist (see page 131)

Syntax 1

```
DISplay:PERSistence {OFF|INFPersist|VARPersist}
```

Syntax 2

```
DISplay:PERSistence?
```

Arguments

- `OFF`

  This disables the persistence aspect of the display.

- `INFPersist`

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

- `VARPersist`

  This sets a display mode where set pixels are gradually dimmed.

Example 1

```
DISplay:PERSistence VARPersist
```

This command sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARPersist command.

Example 2

```
DISplay:PERSistence?
```

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

DISplay:STYle

Description

This command sets or queries how the data is displayed for normal and FastAcq modes. This command is equivalent to setting Display Style from the Display menu.

Group

Display Control

Syntax 1

```
DISplay:STYle {VECtors|DOTs|INTENSIFied}
```

Syntax 2

```
DISplay:STYle?
```

Arguments

- `VECtors`

  This connects adjacent data points. New points immediately replace old ones.

- `DOTs`

  This displays individual data points. New points immediately replace old ones.

- `INTENSIFied`

  This 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.

Example 1

```
DISplay:STYle VECtors
```

This command sets the display to connect adjacent data points.

Example 2

```
DISplay:STYle?
```

This query might return `:DISPLAY:STYLE INTENSIFIED`, indicating that interpolated samples are not displayed.

DISplay:TRIGBar

Description

This command controls or queries the display of the trigger-level indicator bar/s. Indicator bars show where the trigger voltage level is set.

The oscilloscope 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 1

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

Syntax 2

```
DISplay:TRIGBar?
```

Arguments

- `OFF`

  This removes the trigger indicator bar from the display.

- `SHORt`

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

- `LONG`

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

Example 1

```
DISplay:TRIGBar LONG
```

This command sets the display to show a long trigger indicator bar (or bars).

Example 2

```
DISplay:TRIGBar?
```

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

DISplay:TRIGT

Description

This command controls or queries the display of the trigger point indicator. This is equivalent to selecting Objects from the Display menu. The query form returns an ON (1) or an OFF (0).

Group

Display Control

Syntax 1

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

Syntax 2

```
DISplay:TRIGT?
```

Arguments

- `ON`

  This displays a trigger indicator on each of the displayed waveforms. The trigger indicator is in reverse video for the selected waveform.

- `OFF`

  This removes the trigger indicator from the display.

- `<NR1>`

  A 0 removes the trigger indicator from the display; a 1 displays a trigger indicator on each of the displayed waveforms.

Example 1

```
DISplay:TRIGT ON
```

This command sets the display to show a trigger indicators.

Example 2

```
DISplay:TRIGT?
```

This query might return `:DISPLAY:TRIGT 0`, indicating that the trigger indicator is removed from the display.

DISplay:VARPersist

Description

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

Related Commands

DISplay:PERSistence (see page 127)

Syntax 1

```
DISplay:VARPersist <NR3>
```

Syntax 2

```
DISplay:VARPersist?
```

Arguments

- `<NR3>`

  This indicates the persistence decay time setting in numeric seconds. The persistence time ranges from 32 ms to 10 s.

Example 1

```
DISplay:VARPersist 5E-1
```

This command sets the persistence decay time to 500 ms.

Example 2

```
DISplay:VARPersist?
```

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

File System Command Group

File System Overview

You 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

File System Commands

Command	Description
FILESystem?	Returns the file system state
FILESystem:COPy	Copies one or more files to a new file
FILESystem:DELEte	Deletes a named file or directory
FILESystem:DIR?	Returns a list of directory contents
FILESystem:MKDir	Makes a new directory
FILESystem:PRInt	Prints a named file to the named port
FILESystem:READFile	Copies the named file to the GPIB port
FILESystem:REName	Assigns a new name to an existing file
FILESystem:RMDir	Deletes the named directory
FILESystem:WRITEFile	Copies the GPIB port block data to a named file

FILESystem?

Description

This query-only command returns the state of the file system.

Group

File System

Related Commands

FILESystem:COPy (see page 135), FILESystem:DELEte (see page 136), FILESystem:DIR? (see page 137), FILESystem:PRInt (see page 139), FILESystem:READFile (see page 140), FILESystem:REName (see page 141), FILESystem:WRITEFile (see page 143)

Syntax

```
FILESystem?
```

Arguments

None.

Example

```
FILESystem?
```

This query might return `:FILESYSTEM:DIR`, indicating that a list of files and directories exist in the default directory.

FILESystem:COPy

Description

This command (no query form) copies a named file or files to a new file. The new file may be in a totally separate directory than the old file. Also, you can use wild card characters (*.*) to copy multiple files with one command.

Group

File System

Related Commands

FILESystem:DELEte (see page 136)

Syntax

```
FILESystem:COPy {<source file path>,
<destination file path>|<source directory path>,
<destination file path>|<source directory path>,
<destination directory path>}
```

Arguments

- `<file path>`

  This 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. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended. You can also use the inclusive filename *.* in the source file path to copy all files.

- `<directory path>`

  This is a quoted string that defines the directory. Input the directory using the form `<drive>/<dir>/<directory name>.<drive>` and one or more `<dir>`s are optional. `<directory name>` stands for a directory name of up to 128 characters. While filename extensions are not required, they are highly recommended.

Example

```
FILESystem:COPy C:/MYDIR/"TEK00001,SET","C:/TEK00001.SET"
```

This command copies the file named `TEK00001.SET`, located in the `MYDIR` directory on the `C` drive to a file named `TEK00001.SET` in the root directory.

FILESystem:DELEte

Description

This command (no query form) deletes a named file or directory.

Group

File System

Related Commands

FILESystem:COPy (see page 135)

Syntax

```
FILESystem:DELEte <file path>
```

Arguments

- `<file path>`

  This 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 oscilloscope will delete the file in the current directory. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended. You can also use the inclusive filename *.* to delete all files.

Example 1

```
FILESystem:DELEte "NOT_MINE.SET"
```

This command deletes the file named `NOT_MINE.SET` on the default drive and directory.

FILESystem:DIR?

Description

This query-only command returns a list quoted string. Each string contains the name of a file or directory in the current directory.

Group

File System

Related Commands

FILESystem:MKDir (see page 138)

Syntax

```
FILESystem:DIR?
```

Arguments

None.

Example

```
FILESystem:DIR?
```

This query returns a list of files and directories in the default directory.

FILESystem:MKDir

Description

This command (no query form) creates a new directory.

Group

File System

Related Commands

FILESystem:DIR? (see page 137)

Syntax

```
FILESystem:MKDir <directory path>
```

Arguments

- `<directory path>`

  This is a quoted string that defines the directory. Input the directory using the form `<drive>/<dir>/<directory name>`.`<drive>` and one or more `<dir>`s are optional. `<directory name>` stands for a directory name of up to 128 characters. While filename extensions are not required, they are highly recommended.

  The current directory refers to the name of a directory as returned to the FILESystem:CWD? query.

Example

```
FILESystem:MKDir C:/"NEATPICS"
```

This command creates the directory named `NEATPICS` on the current drive.

FILESystem:PRInt

Description

This command (no query form) prints a named file to a named port.

Group

File System

Related Commands

FILESystem:READFile (see page 140)

Syntax

```
FILESystem:PRInt <filepath>, GPIb
```

Arguments

- `<file path>`

  This 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. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended. You can also use the inclusive filename *.* to print all files.

- `GPIb`

  This specifies that the hardcopy is sent out the GPIB port.

Example

```
FILESystem:PRInt C:/"TEK00000.IBM",GPIb
```

This command sends the file named `TEK00000.IBM` out the GPIB port.

FILESystem:READFile

Description

This command (no query form) copies a named file to the GPIB port.

Group

File System

Related Commands

FILESystem:PRInt (see page 139)

Syntax

```
FILESystem:READFile <filepath>
```

Arguments

- `<file path>`

  This 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. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended. You can also use the inclusive filename *.* to read all files.

Example

```
FILESystem:READFile C:/"ABC.DEF"
```

This command reads a file named `"ABC.DEF"` on the C drive and sends it to the GPIB port.

FILESystem:REName

Description

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>
```

Arguments

- `<old file path>`

  This is a quoted string that defines the file to rename. Input the file path using the form `<drive>/<dir>/<filename>`. `<drive>` and one or more `<dir>`s are optional. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended.

- `<new file path>`

  This is a quoted string that defines the new name of the file. Input the file path using the form `<drive>/<dir>/<filename>`. `<drive>` and one or more `<dir>`s are optional. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended.

Example

```
FILESystem:REName C:/"TEK00000.SET",C:/"MYSETTING.SET"
```

This command 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

Description

This command (no query form) deletes a named directory. The directory and all of the files and subdirectories it contains are deleted.

Group

File System

Syntax

```
FILESystem:RMDir <directory path>
```

Arguments

- `<directory path>`

  This is a quoted string that defines the directory. Input the directory using the form `<drive>/<dir>/<directory name>.<drive>` and one or more `<dir>`s are optional. `<directory name>` stands for a directory name of up to 128 characters. While filename extensions are not required, they are highly recommended.

  The current directory refers to the name of a directory as returned to the FILESystem:CWD? query.

Example

```
FILESystem:RMDir C:/"NOT-MINE"
```

This command creates the directory named `NOT-MINE` on the C drive.

FILESystem:WRITEFile

Description

This command (no query form) copies the GPIB port block data to a named file.

Group

File System

Syntax

```
FILESystem:WRITEFile <file path>,<data>
```

Arguments

- `<file path>`

  This 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. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended.

- `<data>`

  This is sent in the indefinite format of ARBITRARY BLOCK PROGRAM DATA encoding as described in IEEE 488.2.

Example

```
FILESystem:WRITEFile C:/"ABC.DEF",#0123<NL+EOL>
```

This command creates a file named `"ABC.DEF"` on the C drive, containing the three bytes (in hex format) 31, 32, and 33.

Hard Copy Command Group

Hard Copy Overview

Hard copy commands enables 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.

Hard Copy Commands

Command	Description
HARDCopy?	Returns format, layout, and port information
HARDCopy	Sends a screen copy that can either be stored or printed
HARDCopy:FILEName?	Returns the hardcopy file path
HARDCopy:FILEName	Sets the hardcopy file path
HARDCopy:PORT?	Returns the data format that will be used on the next hardcopy command
HARDCopy:PORT	Sets whether the data will be sent to a file or printed on the next hardcopy command

HARDCopy

Description

This command sends a copy of the screen display followed by an EOI to the port specified by HARDCopy:PORT. The format and layout of the output is specified with the HARDCopy:FORMat and HARDCopy:LAYout commands. This command is equivalent to pressing the **PRINT** button on the front panel.

The HARDCopy query returns format, layout and port information.

Group

Hardcopy

Related Commands

*WAI (see page 328), *CLS? (see page 318)

Syntax 1

```
HARDCopy STArt
```

Syntax 2

```
HARDCopy?
```

Arguments

- `STARt`

  This initiates a screen copy that is sent to the controller where it can be stored in a file or redirected to a printing device.

  Note: Use the *WAI command between HARDCopy STARt commands to ensure that the first hardcopy is complete before starting another.

Example 1

```
HARDCopy STARt
```

This command initiates a screen copy that can either be stored in file format or printed.

Example 2

```
HARDCopy?
```

This query returns format, layout and port information.

HARDCopy:FILEName

Description

This command sets or returns the file that will be sent hardcopy data on the next HARDCopy command.

Group

Hardcopy

Related Commands

HARDCopy (see page 146)

Syntax 1

```
HARDCopy:FILEName <file path>
```

Syntax 2

```
HARDCopy:FILEName?
```

Argument

- `<file path>`

  This specifies that the hardcopy 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. `<filename>` stands for a filename of up to 128 characters. While filename extensions are not required, they are highly recommended.

  You can automatically create different names for files. You do this by using the question mark (?) as a special wildcard character. These stand for numbers the oscilloscope will insert sequentially in the filename. For example, if you placed two question marks at the end of the filename, then the oscilloscope would append 00 to the first file created, 01 to the next, and 02 to the next. This helps you automatically create different names for files. It is particularly useful in automated testing situations.

Example 1

```
HARDCopy:FILEName "TEK.IBM"
```

This command sets TEK.IBM as the selected file name.

Example 2

```
HARDCopy:FILEName?
```

This query might return `HARDCOPY:FILENAME "TEK.IBM"` as the selected file.

Example 3

```
HARDCopy:FILEName "TEK??.IBM"
```

This command sets TEK as the selected file name with a numeric, two-digit suffix. The oscilloscope might return `HARDCOPY:FILENAME "TEK00.IBM";"TEK01.IBM"`, indicating TEK.00.IBM as the first file and TEK.01.IBM as the second file.

HARDCopy:PORT

Description

This command selects or returns whether the hardcopy data will be sent to a file or printed on the next hardcopy command (i.e. 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

Related Commands

HARDCopy (see page 146), HARDCopy:FILEName (see page 147)

Syntax 1

```
HARDCopy:PORT {FILE|PRINTER}
```

Syntax 2

```
HARDCopy:PORT?
```

Arguments

- `FILE`

  This specifies that the hardcopy is stored in the file specified in the HARDCopy:FILEName command.

- `PRINTER`

  This specifies that the hardcopy is sent to the printer specified in the Print dialog box.

Example 1

```
HARDCopy:PORT FILE
```

This command stores the hardcopy in the file specified in the HARDCopy:FILEName command.

Example 2

```
HARDCopy:PORT?
```

This query might return `HARDCOPY:PORT FILE`, indicating that hardcopies are sent to the specified file.

Histogram Command Group

Histogram Overview

Histogram commands let you select the type of histogram, what part of the waveform should go into the histogram, and histogram statistics. You can 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.

Histogram Commands

Command	Description
HIStogram?	Return all histogram parameters
HIStogram:BOX?	Returns the left, top, right, and bottom positions of the histo box, in source waveform coordinates
HIStogram:BOX	Sets the left, top, right, and bottom positions of the histo box, in source waveform coordinates
HIStogram:BOXPcnt?	Returns same as HIStogram:BOX, but in percentage coordinates, with 0,0 upper left and 100,100 lower right
HIStogram:BOXPcnt	Sets 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:DISplay?	Returns whether histogram data is displayed and the current display type setting
HIStogram:DISplay	Sets whether histogram data is displayed and what display type to use
HIStogram:FUNCTION?	Returns the current histogram type
HIStogram:FUNCtion	Sets the type of histogram to create
HIStogram:MODe?	Returns the current histogram type or that histogram display is disabled
HIStogram:MODe	Sets type of histogram to create or turns histograms off
HIStogram:SIZe?	Returns the width (or height) of the histogram on the screen in divisions
HIStogram:SIZe	Sets the width (or height) of the histogram on the screen in divisions
HIStogram:SOUrce?	Returns source waveform for histogram
HIStogram:SOUrce	Sets source waveform and time base for histogram
HIStogram:STATE?	Returns whether histogram calculations are enabled.
HIStogram:STATE	Enables or disables histogram calculations

HIStogram?

Description

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

Related Commands

MEASUrement:METHod (see page 258)

Syntax 1

```
HIStogram?
```

Arguments

None,

Example 1

```
HIStogram?
```

This query might return :HISTOGRAM:BOXPCNT 3.0000E+01, 2.5100E+01,7.0000E+01,7.5200E+01;DISPLAY LINEAR ;STATE 0;FUNCTION HORIZONTAL;SIZE 2.0000E+00 ;SOURCE CH1

HIStogram:Box

Description

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. This command is similar to the HIStogram:BOXPcnt command except that command uses percentage coordinates to define the boundaries of the histogram box. This command is equivalent to selecting Waveform Histograms from the Measure menu and reviewing the histogram Left, Right, Top, and Bottom Limits.

> **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

Related Commands

HIStogram:BOXPcnt (see page 153), HIStogram:STATE (see page 160)

Syntax 1

```
HIStogram:Box <NR3>,<NR3>,<NR3>,<NR3>
```

Syntax 2

```
HIStogram:Box?
```

Arguments

- `<NR3>`

  This specifies the left position of the histogram box in source waveform coordinates.

- `<NR3>`

  This specifies the top position of the histogram box in source waveform coordinates.

- `<NR3>`

  This specifies the right position of the histogram box in source waveform coordinates.

- `<NR3>`

  This specifies the bottom position of the histogram box in source waveform coordinates.

Example 1

```
HIStogram:Box 1E-9, 0.250, 2E-9, 0.500
```

This command defines the coordinates of the histogram box in source waveform coordinates.

Example 2

```
HIStogram:Box?
```

This query 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

Description

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

Related Commands

HIStogram:Box (see page 152), HIStogram:STATE (see page 160)

Syntax 1

```
HIStogram:BOXPcnt <NR3>, <NR3>, <NR3>, <NR3>
```

Syntax 2

```
HIStogram:BOXPcnt?
```

Arguments

- `<NR3>`

  This specifies the left position of the histogram box in percentage coordinates. The default value is 30.0.

- `<NR3>`

  This specifies the top position of the histogram box in percentage coordinates. The default value is 25.1.

- `<NR3>`

  This specifies the right position of the histogram box in percentage coordinates. The default value is 70.0.

- `<NR3>`

  This specifies the bottom position of the histogram box in percentage coordinates. The default value is 75.2.

Example 1

```
HIStogram:BOXPct 30.0,25.1,70.0,75.2
```

This command sets the hardcopy page-layout format to default values.

Example 2

```
HIStogram:BOXPct?
```

This query might return `:HISTOGRAM:BOXPCT 3.0000E+01,25.1000E+01,7.0000E+01,7.5200E+01`.

HIStogram:COUNt

Description

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

Related Commands

HIStogram:Box (see page 152), HIStogram:BOXPcnt (see page 153)

Syntax

```
HIStogram:COUNt RESET
```

- `RESET`

  This zeros the histogram count.

Example

```
HIStogram:COUNt RESET
```

This command clears the count and statistics for the histogram and the histogram source data.

HIStogram:DISplay

Description

This command sets the format of the histogram displayed or whether the histogram is displayed. The query form of this command either returns the histogram display format or that histogram display is disabled. This command is equivalent to selecting Waveform Histogram from the Measure menu and then ensuring that Display is set to On.

Group

HIStogram

Related Commands

HIStogram:MODe (see page 157), HIStogram:STATE (see page 160)

Syntax 1

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

Syntax 2

```
HIStogram:DISplay?
```

Arguments

- `<OFF>`

  This disables the histogram display but allows the histogram data to accumulate.

- `<LOG>`

  This specifies that the histogram display type is set to Log format.

- `<LINEAr>`

  This sets the histogram display type to Linear format.

Example 1

```
HIStogram:DISplay OFF
```

This command disables the histogram display.

Example 2

```
HIStogram:DISplay?
```

This query might return `:HISTOGRAM:DISPLAY LOG`, indicating that the current histogram type being displayed is in Log format.

HIStogram:FUNCtion

Description

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

Related Commands

HIStogram:MODe (see page 157), HIStogram:STATE (see page 160)

Syntax 1

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

Syntax 2

```
HIStogram:FUNCtion?
```

Arguments

- `HORizontal`

  This displays a horizontally positioned histogram that shows time distribution.

- `VERTical`

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

Example 1

```
HIStogram:FUNCtion HORizontal
```

This command sets the histogram mode to the horizontal display mode.

Example 2

```
HIStogram:FUNCtion?
```

This query might return `:HISTOGRAM:FUNCTION VERTICAL`, indicating that the vertical display mode is selected for the histogram.

HIStogram:MODe

Description

This command selects the type of histogram to create or disables the histogram display. The query form of this command query returns the current histogram type or that histogram display functionality 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 compatibility purposes.

Group

HIStogram

Related Commands

HIStogram:FUNCTION (see page 156), HIStogram:STATE (see page 160), HIStogram:DISplay (see page 155)

Syntax 1

```
HIStogram:MODe {HORizontal|VERTical|OFF}
```

Syntax 2

```
HIStogram:MODe?
```

Arguments

- `HORizontal`

  This enables a horizontally positioned histogram that shows time distribution.

- `VERTical`

  This enables a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).

- `OFF`

  This disables collection of the histogram measurement.

Example 1

```
HIStogram:MODe HORizontal
```

This command sets the type of histogram created to horizontal, which displays a horizontally positioned histogram that shows time distribution.

Example 2

```
HIStogram:MODe?
```

This query might return `:HISTOGRAM:MODE OFF`, indicating that histogram display is disabled.

HIStogram:SIZe

Description

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 Histo Size box.

Group

HIStogram

Related Commands

HIStogram:DISplay (see page 155)

Syntax 1

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

Syntax 2

```
HIStogram:SIZe?
```

Arguments

- `<NR3>`

  This 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.

Example 1

```
HIStogram:SIZe 2.0
```

This command sets the size of the histogram to 2.0 divisions.

Example 2

```
HIStogram:SIZe?
```

This query might return `:HISTOGRAM:SIZE 2.0000E+01`, indicating the histogram size is set to 2.0 divisions.

HIStogram:SOUrce

Description

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

Related Commands

HIStogram:DISplay (see page 155)

Syntax 1

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

Syntax 2

```
HIStogram:SOUrce?
```

Arguments

- `CH<x>`

  This selects a channel waveform as the source for the histogram. The channel waveform range is from 1 through 4.

- `MATH<x>`

  This selects a math waveform as the source for the histogram. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This selects a reference waveform as the source for the histogram. The valid reference waveform range is from 1 through 4.

Example 1

```
HIStogram:SOUrce CH1
```

This command enables the channel 1 waveform to be compared against the histogram box. The default time base, Main, is used for the histogram display.

Example 2

```
HIStogram:SOUrce?
```

This query might return `:HISTOGRAM:SOURCE CH1`, indicating that the waveform for channel 1 is the source for the histogram.

HIStogram:STATE

Description

This command sets or queries whether the histogram calculations are enabled. This is equivalent to selecting Waveform Histograms from the Measure menu.

Group

HIStogram

Related Commands

HIStogram:MODe (see page 157), MEASUrement:METHod (see page 258)

Syntax 1

```
HIStogram:STATE {ON|OFF|NR1}
```

Syntax 2

```
HIStogram:STATE?
```

Arguments

- `<ON>`

  This enables histogram calculations.

- `<OFF>`

  This disables the histogram calculations.

- `<NR1>`

  A 0 disables histogram calculations; any other value enables histogram calculations.

Example 1

```
HIStogram:STATE ON
```

This command enables histogram calculations.

Example 2

```
HIStogram:STATE?
```

This query might return `:HISTOGRAM:STATE 0`, indicating that histogram calculations are currently disabled.

Horizontal Command Group

Horizontal Overiview

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 oscilloscopes.

Horizontal Fastframe (also known as memory segmentation) commands are designed for situations where you want to capture a series of triggered acquisitions with minimal dead time between them. You can use the Horizontal Fastframe commands to do the following:

- Maintain Fastframe acquisitions with time stamp information for channel-based waveforms (in Sample, Peak Defect, and Hi Res modes)
- Display a list of time stamp and basic delta information on a Fastframe waveform, or Fastframe waveforms within a single frame (in either YT or XY format)
- Zoom Fastframe waveforms within a single frame

Horizontal Commands

Command	Description
HORizontal?	Returns all learnable settings for the horizontal commands
HORizontal:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed
HORizontal:FASTframe:COUNt?	Returns FastFrame frame count
HORizontal:FASTframe:COUNt	Sets FastFrame frame count
HORizontal:FASTframe:LENgth?	Returns the horizontal record length per frame
HORizontal:FASTframe:LENgth	Sets the horizontal record length to the number of sample points per frame
HORizontal:FASTframe:REF:FRAme?	Returns the FastFrame reference frame number
HORizontal:FASTframe:REF:FRAme	Sets the FastFrame reference frame number
HORizontal:FASTframe:REF:Source?	Returns the FastFrame Reference waveform source
HORizontal:FASTframe:REF:Source	Sets the FastFrame Reference waveform source
HORizontal:FASTframe:SELECTED :<wfm>?	Returns the FastFrame Selected frame number on the specified waveform
HORizontal:FASTframe:SELECTED :<wfm>	Sets the FastFrame Selected frame number on the specified waveform
HORizontal:FASTframe:STATE?	Returns the state of the FastFrame acquisition
HORizontal:FASTframe:STATE	Sets the state of the FastFrame acquisition
HORizontal:FASTframe:TIMEStamp :All:<wfm>?	Returns the frame number and time stamp for each frame between requested frames within the specified waveform
HORizontal:FASTframe:TIMEStamp :BETWeen:<wfm>?	Returns the relative trigger for the delta time between the specified frames
HORizontal:FASTframe:TIMEStamp :DELTa:<wfm>?	Returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform
HORizontal:FASTframe:TIMEStamp :FRAME:<wfm>?	Returns the absolute trigger date and time for the specified frame and waveform.
HORizontal:FASTframe:TIMEStamp :REF?	Returns the absolute trigger date and time for FastFrame reference
HORizontal:FASTframe:TIMEStamp :SELECTED:<wfm>?	Returns the absolute trigger date and time for FastFrame Selected,

	within the specified waveform
HORizontal:FASTframe:TRACk?	Returns the state of the FastFrame tracking feature
HORizontal:FASTframe:TRACk	Sets the FastFrame tracking feature
HORizontal:MAIn?	Returns the time per division of the main time base
HORizontal[:MAIn]:DELay:POSition?	Returns the main time base position when Horizontal Delay Mode is turned off
HORizontal[:MAIn]:DELay:POSition	Sets the main time base position when Horizontal Delay Mode is turned off
HORizontal[:MAIn]:DELay:MODe?	Returns the main time base trigger delay mode
HORizontal[:MAIn]:DELay:MODe	Sets the main time base trigger delay mode
HORizontal[:MAIn]:DELay:TIMe?	Returns the main time base trigger delay time
HORizontal[:MAIn]:DELay:TIMe	Sets the main time base trigger delay time
HORizontal[:MAIn]:POSition?	Returns the waveform horizontal position on the display
HORizontal[:MAIn]:POSition	Sets the waveform horizontal position on the display
HORizontal:MAIn:SAMPLERate?	Returns the current horizontal sample rate
HORizontal:MAIn:SAMPLERate	Sets the horizontal sample rate to the desired number of samples per second
HORizontal[:MAIn]:SCAle?	Returns the main time base horizontal scale
HORizontal[:MAIn]:SCAle	Sets time per division for the main time base
HORizontal:MAIn:UNIts?	Returns the units for the horizontal main time base
HORizontal:MAIn:UNIts	Sets the units for the horizontal main time base
HORizontal:MAIn:UNIts:OFFSet?	Returns the offset units for the horizontal main time base trigger delay
HORizontal:MAIn:UNIts:OFFSet	Sets the offset units for the horizontal main time base trigger delay
HORizontal:MAIn:UNIts:STRing?	Returns the units string for the horizontal main time base trigger delay
HORizontal:MAIn:UNIts:STRing	Sets the units string for the horizontal main time base trigger delay
HORizontal:MAIn:UNIts:VALue?	Returns the unit value for the horizontal main time base trigger

	delay
HORizontal:MAIn:UNIts:VALue	Sets the unit value for the horizontal main time base trigger delay
HORizontal:POSition?	Returns the waveform's horizontal position on the display
HORizontal:POSition	Positions the waveform horizontally on the display
HORizontal:RECOrdlength?	Returns the current horizontal record length
HORizontal:RECOrdlength	Sets the horizontal record length to the number of data points in each frame
HORizontal:RESOlution?	Returns the current horizontal recordlength
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
HORizontal:ROLL?	Returns the horizontal roll mode status
HORizontal:ROLL	Sets the horizontal roll mode
HORizontal:SCAle?	Returns the time per division for the main time base
HORizontal:SCAle	Sets the time per division for the main time base
HORizontal:TRIGger:POSition?	Returns the position of the horizontal trigger
HORizontal:TRIGger:POSition	Sets the position of the horizontal trigger

HORizontal?

Description

This query-only command returns all learnable settings for the horizontal commands.

Group

HORizontal

Syntax

```
HORizontal?
```

Example

```
HORizontal?
```

This query might return `:HORIZONTAL:DELAY:MODE 1`

```
;POSITION 5.0000E+01;TIME 0.0000E+00;
:HORIZONTAL:MAIN:SCALE 2.0000E-07;POSITION 5.0000E+01
;UNITS:VALUE 1.0000E+00;STRING "s";OFFSET 0.0000E+00;
:HORIZONTAL:RECORDLENGTH 5000;ROLL AUTO;FASTFRAME:STATE 0
;LENGTH 5000;COUNT 2;SELECTED:CH1 2;CH2 2;CH3 2;CH4 2
;MATH1 2;MATH2 2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2
;REF4 2;:HORIZONTAL:FASTFRAME:REF:SOURCE CH1;FRAME 1;
:HORIZONTAL:FASTFRAME:TRACK LIVE
```

HORizontal:DIVisions?

Description

This query-only command returns the number of graticule divisions over which the waveform is displayed.

Group

Horizontal

Syntax

```
HORizontal:DIVisions?
```

Arguments

None.

Example

```
HORizontal:DIVisions?
```

This query might return `:HORIZONTAL:DIVISIONS 1.0000E-01`, indicating that the waveform is displayed across one division.

HORizontal:FASTframe:COUNt

Description

This command sets or queries FastFrame frame count. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and entering a value in the Frame Count box. FastFrame, also known as memory segmentation, captures a series of triggered acquisitions with minimal intervening time. This command is equivalent to selecting Fast Frame Setup from the Horiz/Acq menu and entering a value in the Frame Count box.

Group

Horizontal

Related Commands

HORizontal:RECOrdlength (see page 192)

Syntax 1

```
HORizontal:FASTframe:COUNt <NR1>
```

Syntax 2

```
HORizontal:FASTframe:COUNt?
```

Arguments

- `<NR1>`

  This indicates the number of frames to acquire.

Example 1

```
HORizontal:FASTframe:COUNt 2
```

This command sets up FastFrame mode to acquire two frames (segments) of data.

Example 2

```
HORizontal:FASTframe:COUNt?
```

This query might return `:HORIZONTAL:FASTFRAME:COUNT 4`, indicating that FastFrame mode is currently set to acquire 4 frames (segments) of data.

HORizontal:FASTframe:REF:FRAME

Description

This command sets or queries the FastFrame reference frame number. The Reference Frame number is used to calculate time differences for each frame and displays those differences in the graticule. This command is equivalent to selecting FastFrame Setup in the Horiz/Acq menu and entering a value in the Time Stamps/Frame box.

Group

Horizontal

Related Commands

HORizontal:FASTframe:REF:SOUrce (see page 170), HORizontal:FASTframe:TIMEstamp:DELTa:<wfm>? (see page 175), HORizontal:FASTframe:TIMEstamp:REF? (see page 177), HORizontal:FASTframe:TRACk (see page 179)

Syntax 1

```
HORizontal:FASTframe:REF:FRAME <NR1>
```

Syntax 2

```
HORizontal:FASTframe:REF:FRAME?
```

Argument

- `<NR1>`

  This specifies the FastFrame reference frame.

Example 1

```
HORizontal:FASTframe:REF:FRAME 2
```

This command sets the FastFrame reference frame to 2.

Example 2

```
HORizontal:FASTframe:REF:FRAME?
```

This query might return `:HORIZONTAL:FASTFRAME:REF:FRAME 3`, indicating that the FastFrame reference frame is currently set to 3.

HORizontal:FASTframe:LENgth

Description

This command sets or queries the horizontal record length to the number of sample points in each frame. This command is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and entering a value in the Rec Length box. FastFrame, also known as memory segmentation, captures a series of triggered acquisitions with minimal intervening time between them.

Group

Horizontal

Syntax 1

```
HORizontal:FASTframe:LENgth <NR1>
```

Syntax 2

```
HORizontal:FASTframe:LENgth?
```

Argument

- `<NR1>`

  This represents the supported values for horizontal record lengths, which can range from 500 through 400000. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

Example 1

```
HORizontal:FASTframe:LENgth 5000
```

This command sets the horizontal record length to 5000 sample points in each frame.

Example 2

```
HORizontal:FASTframe:LENgth?
```

This query might return `:HORIZONTAL:FASTFRAME:LENGTH 25000`, indicating that the FastFrame record length is set to 25000 sample points in each frame.

HORizontal:FASTframe:REF:SOUrce

Description

This command sets or queries FastFrame Reference waveform source. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and choosing the reference source.

Group

Horizontal

Related Commands

HORizontal:FASTframe:REF:FRAme (see page 168), HORizontal:FASTframe:TIMEstamp:DELTa:<wfm>? (see page 175), HORizontal:FASTframe:TIMEstamp:REF? (see page 177), HORizontal:FASTframe:TRACk (see page 179)

Syntax 1

```
HORizontal:FASTframe:REF:SOUrce <wfm>
```

Syntax 2

```
HORizontal:FASTframe:REF:SOUrce?
```

Argument

- `<wfm>`

  This specifies the FastFrame Reference waveform source. Valid waveforms include CH<x>, MATH<x>, and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Example 1

```
HORizontal:FASTframe:REF:SOUrce CH2
```

This command sets the horizontal FastFrame waveform reference source to CH2.

Example 2

```
HORizontal:FASTframe:REF:SOUrce?
```

This query might return `:HORIZONTAL:FASTFRAME:REF:SOURCE CH2`, indicating that the FastFrame waveform source is currently set to CH2.

HORizontal:FASTframe:SELECTED:<wfm>

Description

This command sets or returns the FastFrame Selected frame number on the specified waveform. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and then choosing the waveform source and frame. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Related Commands

HORizontal:FASTframe:TRACk (see page 179), HORizontal:FASTframe:TIMEstamp:DELTa:<wfm>? (see page 175), HORizontal:FASTframe:TIMEstamp:REF? (see page 177)

Syntax 1

```
HORizontal:FASTframe:SELECTED:<wfm> <NR1>
```

Syntax 2

```
HORizontal:FASTframe:SELECTED?
```

Argument

- `<NR1>`

  This specifies the selected frame number on the specified waveform.

Example 1

```
HORizontal:FASTframe:SELECTED:CH2 33
```

This command sets the FastFrame Selected frame number on channel 2 to 33.

Example 2

```
HORizontal:FASTframe:SELECTED:CH4?
```

This query might return `:HORIZONTAL:FASTFRAME:SELECTED:CH4 25`, indicating that the FastFrame frame number on channel 4 is set to 25.

Example 3

```
HORizontal:FASTframe:SELECTED?
```

This query might return `:HORIZONTAL:FASTFRAME:SELECTED:CH1 2; CH2 2; CH3 2; CH4 2; MATH1 2; MATH2 2; MATH3 2; MATH4 2; REF1 8; REF2 8; REF3 8; REF4 8`, indicating all waveforms' Fastframe selected frame numbers.

HORizontal:FASTframe:STATE

Description

This command sets or returns the state of FastFrame acquisition. This is equivalent to setting FastFrame On in the FastFrame Setup menu. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal time between them.

The digitizing oscilloscope in FastFrame mode is ready to accept a continuous burst of triggers 400 ms after the controller sends the ACQuire:STATE RUN command.

Group

Horizontal

Related Commands

ACQuire:STATE RUN (see page 26)

Syntax 1

```
HORizontal:FASTframe:STATE {ON|OFF|<NR1>}
```

Syntax 2

```
HORizontal:FASTframe:STATE?
```

Arguments

- `<ON>`

  This turns on horizontal FastFrame.

- `<OFF>`

  This turns off horizontal FastFrame.

- `<NR1>`

  A 0 turns off horizontal FastFrame; any other value turns on horizontal FastFrame.

Example 1

```
HORizontal:FASTframe:STATE ON
```

This command turns on horizontal FastFrame.

Example 2

```
HORizontal:FASTframe:STATE?
```

This query might return `:HORIZONTAL:FASTFRAME:STATE 0`, indicating that the current state of FastFrame is off.

HORizontal:FASTframe:TIMEStamp:ALL:<wfm>?

Description

This query-only command returns the frame number and time stamp for each frame between requested frames, inclusive, within the specified waveform. Argument order is unimportant. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:ALL:<wfm>? <NRF>, <NRF>
```

Arguments

- `<NRF>`

  This specifies the selected "from" frame number within the specified waveform.

- `<NRF>`

  This specifies the selected "to" frame number within the specified waveform.

Example

```
HORizontal:FASTframe:TIMEStamp:ALL:CH2 4,1?
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:ALL:CH1 4,1,"02 Mar 2000 20:10:54.542 037 272 620"`, which is the list of time stamps from frame 1 through frame 4 on channel 1.

HORizontal:FASTframe:TIMEStamp:BETWeen:<wfm>?

Description

This query-only command returns the relative trigger for the delta time between the specified frames, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:BETWeen:<wfm>? <NRF>, <NRF>
```

Arguments

- `<NRF>`

  This specifies the selected "from" frame number within the specified waveform.

- `<NRF>`

  This specifies the selected "to" frame number within the specified waveform.

Example

```
HORizontal:FASTframe:TIMEStamp:BETWeen:CH1? 5,2
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN:CH1 5,2,"00:00:00.010 000 540 624"`, which is the delta of channel 1's Time Stamp frame 5 - Time Stamp frame 2 (TS[5]-TS[2]).

HORizontal:FASTframe:TIMEStamp:DELTa:<wfm>?

Description

This query-only command returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:DELTa:<wfm>?
```

Example

```
HORizontal:FASTframe:TIMEStamp:DELTa:CH1?
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA:CH1 "00:00:00.006 000 306 556"`, which is the delta time stamp of FastFrame Selected Frame on channel 1 - FastFrame Reference Frame (TS[C1 Selected]-TS[Reference]).

HORizontal:FASTframe:TIMEStamp:FRAME:<wfm>?

Description

This query-only command returns the absolute trigger date and time for the specified frame and waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Syntax

```
HORizontal:FASTframe:TIMEStamp:FRAME:<wfm>? <NR1>
```

Argument

- `<NR1>`

  The frame number for which the timestamp will be returned on the specified waveform.

Example

```
HORizontal:FASTframe:TIMEStamp:FRAME:CH1? 10
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:FRAME:CH1 10,"02 Mar 2000 20:10:54.536 036 928 432"`, which is the time stamp of channel 1, frame 10.

HORizontal:FASTframe:TIMEStamp:REF?

Description

This query-only command returns the absolute trigger date and time for FastFrame reference.

Group

Horizontal

Related Commands

HORizontal:FASTframe:REF:FRAme (see page 168), HORizontal:FASTframe:REF:SOUrce (see page 170)

Syntax

```
HORizontal:FASTframe:TIMEStamp:REF?
```

Example

```
HORizontal:FASTframe:TIMEStamp:REF?
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:REF "02 Mar 2000 20:10:54.533 036 838 784"`, which is the time stamp of FastFrame Reference.

HORizontal:FASTframe:TIMEStamp:SELECTED:<wfm>?

Description

This query-only command returns the absolute trigger date and time for the FastFrame Selected, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>; the x variable can be expressed as an integer ranging from 1 through 4.

Group

Horizontal

Related Commands

HORizontal:FASTframe:SELECTED:<wfm>? (see page 171)

Syntax

```
HORizontal:FASTframe:TIMEStamp:SELECTED:<wfm>?
```

Example

```
HORizontal:FASTframe:TIMEStamp:SELECTED:REF1?
```

This query might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED:REF1 "02 MAR 2000 20:10:54:539 037 145 340"`, which is the time stamp of FastFrame Selected on reference 1.

HORizontal:FASTframe:TRACk

Description

This command sets up or returns the state of FastFrame tracking feature. This command is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and then clicking the desired Frame Tracking state.

When FastFrame Track is set to "live", the channel and math waveforms are locked together. Adjusting a channel waveform also adjusts a related math waveform. All reference waveforms are also locked together but they are separate from channel and math waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3 and Math4 are also set to Frame 3.

When you set the Selected Frame Source Ref1 to Frame 2, then Selected Frame Ref2, Ref3 and Ref4 are also set to Frame 2. If the Frame Tracking is set to Live, changing Selected Frame Ch1 will not affect the Selected Frame Ref1 frame of the Reference Frame setting.

When FastFrame Track is set to "all", the channel, math and reference waveforms are locked together. Adjusting a channel waveform also adjusts the related math and reference waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3 and Ref4 are also set to Frame 3.

Group

Horizontal

Related Commands

HORizontal:FASTframe:REF:FRAme (see page 168), HORizontal:FASTframe:REF:SOUrce (see page 170), HORizontal:FASTframe:SELECTED:<wfm>? (see page 171)

Syntax 1

```
HORizontal:FASTframe:TRACk {LIVE|ALL}
```

Syntax 2

```
HORizontal:FASTframe:TRACk?
```

Arguments

- `LIVE`

  This sets FastFrame Track to Live.

- `ALL`

  This sets FastFrame Track to All.

Example 1

```
HORizontal:FASTframe:TRACk LIVE
```

This command sets FastFrame Track to live.

Example 2

```
HORizontal:FASTframe:TRACk?
```

This query might return `:HORIZONTAL:FASTFRAME:TRACK ALL`, indicating that all FastFrame Selected Frames and the FastFrame Reference will be locked together.

HORizontal:MAIn?

Description

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

Related Commands

HORizontal:SCAle (see page 195)

Syntax

```
HORizontal:MAIn?
```

Example

```
HORizontal:MAIn?
```

This query might return `:HORIZONTAL:MAIN:SCALE 1.0000E-04;POSITION 5.0000E+01;UNITS:VALUE 1.0000E+00;STRING "s";OFFSET 0.0000E+01`

HORizontal[:MAIn]:POSition

This command either positions the waveform horizontally on the display or returns the horizontal position on the display and is identical to the HORizontal:POSition command. When Horizontal Delay mode is turned off, this command is equivalent to adjusting the **HORIZONTAL POSITION** knob on the front-panel. When Horizontal Delay mode is turned on, this command is equivalent to selecting Horizontal/Acquisition from the Horiz/Acq menu and then choosing a Horizontal Ref Point value.

Group

Horizontal

Related Commands

HORizontal[:MAIn]:DELay:MODe (see page 184)

Syntax 1

```
HORizontal:MAIn:POSition <NR3>
```

Syntax 2

```
HORizontal:MAIn:POSition?
```

Arguments

- `<NR3>`

  This is from 0 to ≈100 and is the percent 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).

Example 1

```
HORizontal:MAIn:POSition 5.000E+01
```

This command sets the horizontal position of the waveform such that 10 percent of the waveform is to the left of screen center.

Example 2

```
HORizontal:MAIn:POSition?
```

This query 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]:SCAle

Description

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

Related Commands

HORizontal:SCAle (see page 182)

Syntax 1

```
HORizontal:MAIn:SCAle <NR3>
```

Syntax 2

```
HORizontal:MAIn:SCAle?
```

Argument

- `<NR3>`

  This is the time per division. The range is from 200 ps through 40 s.

Example 1

```
HORizontal:MAIn:SCAle 2E-6
```

This command sets the main scale to 2µs per division.

Example 2

```
HORizontal:MAIn:SCAle?
```

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

HORizontal[:MAIn]:DELay:POSition

Description

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

Related Commands

HORizontal[:MAIn}:DELay:TIMe (see page 183)

Syntax 1

```
HORizontal[:MAIn]:DELay:POSition <NR3>
```

Syntax 2

```
HORizontal[:MAIn]:DELay:POSition?
```

Arguments

- `<NR3>`

  This is from 0 to ≈100 and is the percent 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).

Example 1

```
HORizontal:DELay:POSition 5E+1
```

This command sets the time base trigger delay position to 50 percent.

Example 2

```
HORizontal:DELay:POSition?
```

This query 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:MODe

Description

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

Related Commands

HORizontal[:MAIn]:DELay:TIMe (see page 185)

Syntax 1

```
HORizontal[:MAIn]:DELay:MODe [ON|OFF|<NR1>]
```

Syntax 2

```
HORizontal[:MAIn]:DELay:MODe?
```

Arguments

- `ON`

  This enables the time base trigger delay mode.

- `OFF`

  This disables the time base trigger delay mode.

- `<NR1>`

  A 0 disables the time base trigger delay mode, any other value enables the time base trigger delay mode.

Example 1

```
HORizontal:DELay:MODe ON
```

This command enables the time base trigger delay mode.

Example 2

```
HORizontal:DELay:MODe?
```

This query might return `:HORIZONTAL:DELAY:MODE 1`, indicating that the time base trigger delay mode is currently enabled.

HORizontal[:MAIn]:DELay:TIMe

Description

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

Related Commands

HORizontal[:MAIn]:DELay:MODe (see page 184), HORizontal[:MAIn]:DELay:POSition (see page 183)

Syntax 1

```
HORizontal[:MAIn]:DELay:TIMe <NR3>
```

Syntax 2

```
HORizontal[:MAIn]:DELay:TIMe?
```

Argument

- `<NR3>`

  This specifies the time base trigger delay time setting, typically represented in seconds.

Example 1

```
HORizontal:DELay:TIMe 5.0E-3
```

This command sets the time base trigger delay time to 5 ms.

Example 2

```
HORizontal:DELay:TIMe?
```

This query might return `:HORIZONTAL:MAIN:DELAY 5.0000E-05`, indicating that the time delay setting is 5 µs.

HORizontal:MAIn:SAMPLERate

Description

This command sets the horizontal sample rate to the desired number of samples per second. The record length is automatically adjusted at the same time to maintain a constant number of data points in each frame. 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

Related Commands

HORizontal:RESOlution (see page 193)

Syntax 1

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

Syntax 2

```
HORizontal:MAIn:SAMPLERate?
```

Arguments

- `<NR3>`

  This represents the size of the sample rate, which ranges from 5 S/s to 250 GS/s.

Example 1

```
HORizontal:MAIn:SAMPLERate 125E6
```

This command sets the sample rate to the sample rate increment that is closest to 125 MS/s.

Example 2

```
HORizontal:MAIn:SAMPLERate?
```

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

HORizontal:MAIn:UNIts

Description

This command sets or returns the units for the horizontal time base. It is equivalent to separately setting the following commands:

- HORizontal:MAIn:UNIts:VALue
- HORizontal:MAIn:UNIts:STRing
- HORizontal:MAIn:UNIts:OFFSet

Group

Horizontal

Related Commands

HORizontal:MAIn:UNIts:VALue (see page 190), HORizontal:MAIn:UNIts:STRing (see page 189), HORizontal:MAIn:UNIts:OFFSet (see page 188)

Syntax 1

```
HORizontal:MAIn:UNIts <NR3> [,<STRing>] [,<NR3>]
```

Syntax 2

```
HORizontal:MAIn:UNIts?
```

Arguments

- `<NR3>`

  This is the time base units multiplier. The instrument scales time/div using this multiplier.

- `<STRing>`

  This is the time base units string.

- `<NR3>`

  This is the time base units offset.

Example 1

```
HORizontal:MAIn:UNIts 5.5,"lum", 2.9
```

This command sets the time base multiplier to 5.5, the units to "lum" and the time base units offset to 2.9.

Example 2

```
HORizontal:MAIn:UNIts?
```

This query might return `:HORIZONTAL:MAIN:UNITS:VALUE 5.5000E+00;STRING "lum", OFFSET 2.9000E+00`

HORizontal:MAIn:UNIts:OFFset

Description

This command sets or returns the offset units for the horizontal time base trigger delay.

Group

Horizontal

Related Commands

HORizontal:MAIn:UNIts (see page 187)

Syntax 1

```
HORizontal:MAIn:UNIts:OFFSet <NR3>
```

Syntax 2

```
HORizontal:MAIn:UNIts:OFFSet?
```

Argument

- `<NR3>`

  This is the time base unit offset.

Example 1

```
HORizontal:MAIn:UNIts:OFFSet 2.9
```

This command sets the horizontal time base trigger delay offset units to 2.9 s.

Example 2

```
HORizontal:MAIn:UNIts:OFFSet?
```

This query might return `:HORIZONTAL:MAIN:UNITS:OFFSET 2.9000E+00`, indicating that the offset units for the horizontal time base trigger delay is set to 2.9 s.

HORizontal:MAIn:UNIts:STRing

Description

This command sets or returns the units string for the horizontal time base trigger delay.

Group

Horizontal

Related Commands

HORizontal:MAIn:UNIts (see page 187)

Syntax 1

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

Syntax 2

```
HORizontal:MAIn:UNIts:STRing?
```

Argument

- `<STRing>`

  This is the time base units string.

Example 1

```
HORizontal:MAIn:UNIts:STRing lum
```

This command sets the units string for the time base trigger delay to lumens.

Example 2

```
HORizontal:MAIn:UNIts:STRing?
```

This query might return `:HORIZONTAL:MAIN:UNITS:STRING "lum"`, indicating that the units string for the time base trigger delay is set to lumens.

HORizontal:MAIn:UNIts:VALue

Description

This command sets or returns the unit value for the horizontal time base trigger delay.

Group

Horizontal

Related Commands

HORizontal:MAIn:UNIts (see page 187)

Syntax 1

```
HORizontal:MAIn:UNIts:VALue <NR3>
```

Syntax 2

```
HORizontal:MAIn:UNIts:VALue?
```

Argument

- `<NR3>`

  This is the time base unit value.

Example 1

```
HORizontal:MAIn:UNIts:VALue 5.5
```

This command sets the unit value for the time base trigger delay to 5.5.

Example 2

```
HORizontal:MAIn:UNIts:VALue?
```

This query might return `:HORIZONTAL:MAIN:UNITS:VALUE 5.5000E+00`, indicating that the unit value for the time base trigger delay is set to 5.5.

HORizontal:POSition

Description

This command either positions the waveform horizontally on the display or returns its horizontal position on the display. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the **HORIZONTAL POSITION** knob on the front panel.

Group

Horizontal

Related Commands

HORizontal[:MAIn]:DELay:MODe

Syntax 1

```
HORizontal:POSition <NR3>
```

Syntax 2

```
HORizontal:POSition?
```

Argument

- `<NR3>`

  This is from 0 to ≈100 and is the percent 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).

Example 1

```
HORizontal:POSition 10
```

This command sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

Example 2

```
HORizontal:POSition?
```

This query might return `:HORIZONTAL:POSITION 5.0000E+01`, indicating the horizontal position of the waveform on the screen.

HORizontal:RECOrdlength

Description

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 1

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

Syntax 2

```
HORizontal:RECOrdlength?
```

Argument

- `<NR1>`

  This represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

Example 1

```
HORizontal:RECOrdlength 50000
```

This command specifies that 50000 data points will be acquired for each record.

Example 2

```
HORizontal:RECOrdlength?
```

This query might return `:HORIZONTAL:RECOrdlength 5000`, indicating that the horizontal record length is equal to 5000 data points.

HORizontal:RESOlution

Description

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 1

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

Syntax 2

```
HORizontal:RESOlution?
```

Argument

- `<NR1>`

  This represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

Example 1

```
HORizontal:RESOlution 50000
```

This command specifies that 50000 data points will be acquired for each record.

Example 2

```
HORizontal:RESOlution?
```

This query might return `:HORIZONTAL:RESOLUTION 5000`, indicating that the horizontal record length is equal to 5000 data points.

HORizontal:ROLL

Description

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 then choosing the desired Roll Mode.

Group

Horizontal

Syntax 1

```
HORizontal:ROLL {AUTO|OFF|ON}
```

Syntax 2

```
HORizontal:ROLL?
```

Arguments

- `AUTO`

  This enables Roll Mode, if the time/division is set appropriately.

- `OFF`

  This disables Roll Mode.

- `ON`

  This enables Roll Mode, if the time/division is set appropriately.

Example 1

```
HORizontal:ROLL ON
```

This command enables Roll Mode.

Example 2

```
HORizontal:RECOrdlength?
```

This query might return `:HORIZONTAL:ROLL OFF`, indicating that the roll mode is disabled.

HORizontal:SCAle

Description

This command sets or queries the time per division for the time base and is identical to the HORizontal:MAIn:SCAle command. It is provided to maintain program compatibility with some older models of Tektronix oscilloscopes. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then entering the desired Scale value.

Group

Horizontal

Related Commands

HORizontal[:MAIn]:SCAle (see page 182)

Syntax 1

```
HORizontal:SCAle <NR3>
```

Syntax 2

```
HORizontal:SCAle?
```

Argument

- `<NR3>`

  This is the time per division. The range is from 200 ps to 40 s.

Example 1

```
HORizontal:SCAle 2E-6
```

This command sets the main scale to 2 µs per division.

Example 2

```
HORizontal:SCAle?
```

This query might return `:HORIZONTAL:SCALE 2.0000E-06`, indicating that the time per division is currently set to 2 µs,

HORizontal:TRIGger:POSition

Description

This command sets or queries the position of the trigger and is identical to the HORizontal:MAIn:POSition command. It is provided to maintain program compatibility with some older models of Tektronix oscilloscopes. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then entering the desired Position.

Group

Horizontal

Related Commands

HORizontal:MAIn:POSition (see page 181)

Syntax 1

```
HORizontal:TRIGger:POSition <NR1>
```

Syntax 2

```
HORizontal:TRIGger:POSition?
```

Argument

- `<NR1>`

  This is the amount of pre-trigger information in the waveform. The range is from 0 to 100%.

Example 1

```
HORizontal:TRIGger:POSition 50
```

This command sets the pre-trigger information in the waveform to 50%.

Example 2

```
HORizontal:TRIGger:POSition?
```

This query might return `:HORIZONTAL:TRIGGER:POSITION 50`

Math Command Group

Math Overview

You 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. 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 not displayed. When you stop either the waveform or the trigger, the math no longer calculates but 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.

Math Commands

Command	Description
MATH<x>?	Returns the specified math waveform settings
MATH<x>:DEFine?	Returns the math<x> definition for the specified waveform
MATH<x>:DEFine	Sets the math<x> waveform definition
MATH<x>:LABEL:NAMe?	Returns the label string used for annotating the displayed math waveform
MATH<x>:LABEL:NAMe	Sets the label string used for annotating the displayed math waveform
MATH<x>:LABEL:XPOS?	Returns the X screen offset where the math waveform label is displayed
MATH<x>:LABEL:XPOS	Sets the X screen offset where the math waveform label is displayed
MATH<x>:LABEL:YPOS?	Returns the Y screen offset where the math waveform label is displayed
MATH<x>:LABEL:YPOS	Sets the Y screen offset where the math waveform label is displayed
MATH<x>:NUMavg?	Returns the acquisition number at which the averaging algorithm will begin exponential averaging
MATH<x>:NUMavg	Sets the acquisition number at which the averaging algorithm will begin exponential averaging
MATH<x>:POSition?	Returns the specified math waveform vertical position
MATH<x>:POSition	Sets the specified math waveform vertical position
MATH<x>:SCAle?	Returns the specified math waveform vertical scale (per div)
MATH<x>:SCAle	Sets the specified math waveform vertical scale (per div)
MATH<x>:SPECTral?	Returns the specified math waveform spectral setups
MATH<x>:SPECTral:CENTER?	Returns the center frequency of the spectral analyzer output data span
MATH<x>:SPECTral:CENTER	Sets the center frequency of the spectral analyzer output data span
MATH<x>:SPECTral:GATEPOS?	Returns the position of the center of the gate
MATH<x>:SPECTral:GATEPOS	Sets the position of the center of the gate
MATH<x>:SPECTral:GATEWIDTH?	Returns the gate width input to the spectral analyzer
MATH<x>:SPECTral:GATEWIDTH	Sets the gate width input to the spectral analyzer
MATH<x>:SPECTral:LOCk?	Returns the state of spectral locking for the specified math waveform
MATH<x>:SPECTral:LOCk	Sets the state of the math spectral locking for the specified math waveform

MATH<x>:SPECTral:MAG?	Returns the units of the SpectralMag function in the specified math string
MATH<x>:SPECTral:MAG	Sets the units of the SpectralMag function in the specified math string
MATH<x>:SPECTral:PHASE?	Returns the units of the SpectralPhase function in the specified math string
MATH<x>:SPECTral:PHASE	Sets the units of the SpectralPhase function in the specified math string
MATH<x>:SPECTral:REFLEVEL?	Returns the vertical position of the output data from the spectral analyzer
MATH<x>:SPECTral:REFLEVEL	Sets the vertical position of the output data from the spectral analyzer
MATH<x>:SPECTral:REFLEVELOffset?	Returns the spectral reference level offset used for calculating the dB value
MATH<x>:SPECTral REFLEVELOffset	Sets the spectral reference level offset used for calculating the dB value
MATH<x>:SPECTral:RESBw?	Returns the resolution bandwidth of the spectral analyzer
MATH<x>:SPECTral:RESBw	Sets the resolution bandwidth of the spectral analyzer
MATH<x>:SPECTral:SPAN?	Returns the specified math waveform's current span value
MATH<x>:SPECTral:SPAN	Sets the top of the span to a value that is closest to the specified value
MATH<x>:SPECTral:SUPPress?	Returns the phase suppression threshold
MATH<x>:SPECTral:SUPPress	Sets the phase suppression threshold
MATH<x>:SPECTral:UNWRap?	Returns whether or not phase unwrap of the spectral analyzer output data is enabled
MATH<x>:SPECTral:UNWRap	Enables or disables phase unwrap of the spectral analyzer output data
MATH<x>:SPECTral:WINdow?	Returns the window function used to multiply the input data to the spectral analyzer
MATH<x>:SPECTral:WINdow	Sets the window function used to multiply the input data to the spectral analyzer

MATH<x>?

Description

This query-only command returns the definition for the math waveform specified by <x>, which ranges from 1 through 4.

Group

Math

Related Commands

SELect:<wfm> (see page 460)

Syntax

```
MATH<x>?
```

Example

```
MATH1?
```

This query might return

```
:MATH1 DEFINE "PROBE POINT7"
;NUMAVG 2;SCALE 1.0000E+00;POSITION 0.0000E+00
;LABEL:NAME "Math1";XPOS 5;YPOS 65;;MATH1:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS 0.0000E+00;GATEWIDTH 1.9996E-06
;REFLEVEL 2.0000E+01;REFLEVELOFFSET 2.2360E-01
;SPAN 1.2500+09;CENTER 6.2500E+08;RESBW 1.0002E+06
;WINDOW GAUSSIAN;SUPPRESS -3.5000EE+01;UNWRAP 0;LOCK 0
```

MATH<x>:DEFIne

Description

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 Ch 1, 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, functions, and operands. As an example, you can enter the expression Log(Ch 1+Ch 2), 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 information about constructing mathematical expressions, see *Creating and Using Math Waveforms* in the User's Guide for this oscilloscope.

Group

Math

Syntax 1

```
MATH<x>:DEFIne <QString>
```

Syntax 2

```
MATH<x>:DEFIne?
```

Argument

- `<QString>`

  This quoted string argument is the mathematical expression that defines the waveform.

Example 1

```
MATH2:DEFIne "Ch1+Ch2"
```

This command adds the Ch 1 waveform and Ch 2 waveforms, storing the results in Math 2.

Example 2

```
MATH2:DEFIne?
```

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

MATH<x>:LABEL:NAMe

Description

This command sets or returns the label string, which is used for annotating the math waveform on the screen. This command is equivalent to selecting Math Setup from the Math menu and then entering a label in the Label box.

Group

Math

Syntax 1

```
MATH<x>:LABEL:NAMe <string>
```

Syntax 2

```
MATH<x>:LABEL:NAMe?
```

Argument

- `<string>`

  This specifies the label to annotate the math waveform.

Example 1

```
MATH2:LABEL:NAMe "Probe point7"
```

This command assigns Probe point7 as the label that annotates Math2.

Example 2

```
MATH2:LABEL:NAMe?
```

This query might return `:MATH2:LABEL:NAME "Probe point7"` as the label that annotates Math 2.

MATH<x>:LABEL:XPOS

Description

This command sets or queries the X screen offset at which the label attached to the displayed 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

Related Commands

MATH<x>:LABEL:YPOS (see page 204)

Syntax 1

```
MATH<x>:LABEL:XPOS <NR1>
```

Syntax 2

```
MATH<x>:LABEL:XPOS?
```

Argument

- `<NR1>`

  This is the location (in pixels) where the waveform label for the selected math is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 to 500.

Example 1

```
MATH2:LABEL:XPOS 50
```

This command moves the waveform label for the MATH3 waveform, so that it begins 50 pixels to the right of the screen's left edge.

Example 2

```
MATH2:LABEL:XPOS?
```

This query might return `:MATH2:LABEL:XPOS 50`, indicating that the waveform label for the MATH2 waveform is currently 50 pixels to the right of the screen's left edge.

MATH<x>:LABEL:YPOS

Description

This command sets or queries the Y screen offset at which the label attached to the displayed math waveform is displayed, relative to the top 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 Y Position box.

Group

Math

Related Commands

MATH<x>:LABEL:XPOS (see page 203)

Syntax 1

```
MATH<x>:LABEL:YPOS <NR1>
```

Syntax 2

```
MATH<x>:LABEL:YPOS?
```

Argument

- `<NR1>`

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

Example 1

```
MATH2:LABEL:YPOS -25
```

This command moves the waveform label for the MATH2 waveform to just beneath (25 pixels below) the top of the screen.

Example 2

```
MATH2:LABEL:YPOS?
```

This query 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

Description

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. If so, it affects all AVG() functions in this math. 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

Related Commands

ACQuire:NUMAVg (see page 23)

Syntax 1

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

Syntax 2

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

Argument

- `<NR1>`

  This specifies the number of acquisitions over which exponential averaging is performed.

Example 1

```
MATH2:NUMAVg 10
```

This command successively averages Math 2 by 10 times.

Example 2

```
MATH2:NUMAVg?
```

This query might return `:MATH2:NUMAVG 10`, indicating 10 Math 2 waveforms are successively averaged before a single acquisition occurs.

MATH<x>:POSition

Description

This command sets or queries the vertical position of the specified Math waveform. The Math waveform is specified by x, which can range 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 channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal 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.

Group

Math, Vertical

Related Commands

CH<x>POSition (see page 437), REF<x>:POSition (see page 457)

Syntax 1

```
MATH<x>:POSition <NR3>
```

Syntax 2

```
MATH<x>:POSition?
```

Argument

- `<NR3>`

  This is the desired position in divisions from the center graticule.

Example 1

```
MATH2:POSition 1.3E+00
```

This command positions the Math 2 input signal 1.3 divisions higher than a position of 0.

Example 2

```
MATH1:POSition?
```

This query might return `:MATH1:POSITION -1.3000E+00`, indicating that the current position of Math 1 is 1.3 divisions below the center graticule.

MATH<x>:SCAle

Description

This command sets or queries the vertical scale of the specified math waveform. The math waveform is specified by x, which can range 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.

In addition to using the MATH<x>:SCAle command, autoscaling occurs when a math is first defined and enabled, or when a math string changes. After the math is calculated for the first time, the instrument determines the min + max of that waveform data. Then, the math position is set 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.

Group

Math, Vertical

Related Commands

CH<x>:SCAle (see page 449), REF<x>:SCAle (see page 458)

Syntax 1

```
MATH<x>:SCAle <NR3>
```

Syntax 2

```
MATH<x>:SCAle?
```

Argument

- `<NR3>`

  This is the scale, in volts, amps or watts per division. The range is from 100.0E-36 through 100.0E+36.

Example 1

```
MATH4:SCAle 100E-03
```

This command sets the Math 4 scale to 100 mV per division.

Example 2

```
CH2:SCAle?
```

This query might return `:MATH2:SCALE 1.0000E+00`, indicating that the current scale setting of Math 2 is 1 volt per division.

MATH<x>:SPECTral?

Description

This query-only command returns the current spectral setups for the specified math waveform. 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?
```

Example

```
MATH1:SPECTral?
```

This query might return
```
:MATH1:SPECTRAL:MAG DB;PHASE DEGREES
;GATEPOS 0.0000E+00;GATEWIDTH 1.9996E-06;REFLEVEL 4.4587+01
;REFLEVELOFFSET 2.2360E-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

Description

This command specifies or returns the center frequency of the spectral analyzer output data span. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Center Freq value.

Group

Math

Related Commands

MATH<x>:SPECTral:SPAN (see page 218)

Syntax 1

```
MATH<x>:SPECTral:CENTER <NR3>
```

Syntax 2

```
MATH<x>:SPECTral:CENTER?
```

Argument

- `<NR3>`

  This is the desired frequency of the spectral analyzer output data span. Units are represented in hertz.

Example 1

```
MATH3:SPECTral:CENTER 10.09E6
```

This command sets the center frequency to the closest value it can attain to 10.09 MHz.

Example 2

```
MATH2:SPECTral:CENTER?
```

This query might return `:MATH2:SPECTral:CENTER 10.0900E+06`, indicating that the center frequency is currently set at 10.09 MHz.

MATH<x>:SPECTral:GATEPOS

Description

This command sets or returns the position of the center of the gate, which is used as the data input to the spectral analyzer. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Gate Pos value.

Group

Math

Related Commands

MATH<x>:SPECTral:GATEWIDTH (see page 211)

Syntax 1

```
MATH<x>:SPECTral:GATEPOS <NRf>
```

Syntax 2

```
MATH<x>:SPECTral:GATEPOS?
```

Argument

- `<NRf>`

  This is the gate position. Units are represented in seconds, with respect to trigger position.

Example 1

```
MATH1:SPECTral:GATEPOS 0
```

This command specifies the position of the center edge of the gate used as the data input to the spectral analyzer.

Example 2

```
MATH2:SPECTral:GATEPOS?
```

This query might return `:MATH2:SPECTRAL:GATEPOS 0.0000E+00`

MATH<x>:SPECTral:GATEWIDTH

Description

This command sets or returns the gate width input, in seconds, to the spectral analyzer. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a value in the Duration box. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Gate Dur value.

Group

Math

Related Commands

MATH<x>:SPECTral:GATEPOS (see page 210)

Syntax 1

```
MATH<x>:SPECTral:GATEWIDTH <NR3>
```

Syntax 2

```
MATH<x>:SPECTral:GATEWIDTH?
```

Argument

- `<NR3>`

  This is the time across the 10-division screen. Units are represented in seconds.

Example 1

```
MATH1:SPECTral:GATEWIDTH 1.0E-3
```

This command sets the gate width input to the spectral analyzer at 1 ms.

Example 2

```
MATH3:SPECTral:GATEWIDTH?
```

This query 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

Description

This command locks menus for Math<x> and Math<y> 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. However, applying spectral locking functionality from the interface is limited to locking Math1 and Math2 and/or Math 3 and Math4.

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 Math 2 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 affect since there is no Math5 to which it can be locked.

Group

Math

Syntax 1

```
MATH<x>:SPECTral:LOCk {ON|OFF}
```

Syntax 2

```
MATH<x>:SPECTral:LOCk?
```

Arguments

- `ON`

  This turns on the parameter lock for the specified math waveform.

- `OFF`

  This turns off the parameter lock for the specified math waveform.

- `<NR1>`

  A 0 disables the parameter lock for the specified math waveform; any other value enables the parameter lock.

Example 1

```
MATH1:SPECTral:LOCk ON
```

This command turns on the parameter Lock1, which causes the parameter for Math1 and Math2 to be locked together.

Example 2

```
MATH1:SPECTral:LOCk?
```

This query might return `:MATH1:SPECTRAL:LOCK 0`, indicating that the parameter Lock1 is turned off.

MATH<x>:SPECTral:MAG

Description

This command controls or returns the units of a SpectralMag function in the math string. This command is equal to selecting Spectral Setup from the Math menu, choosing the Mag tab and then clicking the desired Scale button.

Group

Math

Related Commands

MATH<x>:SPECTral:PHASE (see page 214)

Syntax 1

```
MATH<x>:SPECTral:MAG {LINEAR|DB|DBM}
```

Syntax 2

```
MATH<x>:SPECTral:MAG?
```

Arguments

- `LINEAR`

  This sets the SpectralMag units to linear.

- `DB`

  This sets the SpectralMag units to decibels.

- `DBM`

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

Example 1

```
MATH2:SPECTral:MAG DB
```

This command sets the SpectralMag units for Math2 to decibels.

Example 2

```
MATH2:SPECTral:MAG?
```

This query might return `:MATH2:SPECTRAL:MAG DB`, indicating that the SpectralMag units for Math2 are set to decibels.

MATH<x>:SPECTral:PHASE

Description

This command controls or returns the units of a SpectralPhase function in the math string. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then clicking the desired Scale button.

Group

Math

Related Commands

MATH<x>:SPECTral:MAG (see page 213)

Syntax 1

```
MATH<x>:SPECTral:PHASE {DEGREES|RADIANS|GROUPDELAY}
```

Syntax 2

```
MATH<x>:SPECTral:PHASE?
```

Arguments

- `DEGREES`

  This sets the SpectralPhase units to degrees.

- `RADIANS`

  This sets the SpectralPhase units to radians.

- `GROUPDELAY`

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

Example 1

```
MATH2:SPECTral:PHASE DEGREES
```

This command sets the SpectralPhase units for Math2 to degrees.

Example 2

```
MATH2:SPECTral:PHASE?
```

This query might return `:MATH2:SPECTRAL:PHASE RADIANS`, indicating that the SpectralPhase units for Math2 are set to radians.

MATH<x>:SPECTral:REFLevel

Description

This command specifies the vertical position of the output data from the spectral analyzer on the display screen. The numerical value represents the position at the top of the display graticule. 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

Related Commands

MATH<x>:SPECTral:REFLEVELOffset (see page 216), MATH<x>:SCAle (see page 207), MATH<x>:POSition (see page 206)

Syntax 1

```
MATH<x>:SPECTral:REFLevel <NR3>
```

Syntax 2

```
MATH<x>:SPECTral:REFLevel?
```

Argument

- `<NR3>`

  This is the position that represents the top of the display screen graticule. The range depends on the units and both the MATH<x>:SCAle and MATH<x>:POSition settings.

Example 1

```
MATH1:SPECTral:REFLevel -10
```

This 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.

Example 2

```
MATH1:SPECTral:REFLevel?
```

This query 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

Description

This command sets or returns the spectral level offset used for calculating the dB value. 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

Related Commands

MATH<x>:SPECTral:REFLevel (see page 215)

Syntax 1

```
MATH<x>:SPECTral:REFLEVELOffset {DBM|<NR3>}
```

Syntax 2

```
MATH<x>:SPECTral:REFLEVELOffset?
```

Arguments

- `DBM`

  This specifies the reference level used for calculation to be equivalent to 1 mW into 50 ohms (Zero dB will occur at this level).

- `<NR3>`

  This specifies the reference level used for calculation of the decibel value when the output units are Log.

Example 1

```
MATH1:SPECTral:REFLEVELOFFset 0.5
```

This sets the reference level for the Log calculation for decibel to be 0.5. dB = A x Log(y/<NR3>) where A is 10 if the input units are watts and A is 20 if the input units are otherwise.

Example 2

```
MATH1:SPECTral:REFLEVELOffset DBM
```

This sets the decibel reference to be equivalent to 1 mW into 50 ohms. 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.

Example 3

```
MATH1:SPECTral:REFLEVELOffset?
```

This query might return `:MATH1:SPECTRAL:REFLEVELOFFSET 2.23360E-01`, indicating that the spectral reference level offset is 223.6 mV.

MATH<x>:SPECTral:RESBw

Description

This command sets or returns the resolution bandwidth of the spectral analyzer. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a value in the Res BW box.

Group

Math

Related Commands

MATH<x>:SPECTral:GATEWIDTH (see page 211), MATH<x>:SPECTral:CENTER (see page 209), MATH<x>:SPECTral:SPAN (see page 218), MATH<x>:SPECTral:WINdow (see page 221)

Syntax 1

```
MATH<x>:SPECTral:RESBw <NR3>
```

Syntax 2

```
MATH<x>:SPECTral:RESBw?
```

Argument

- `<NR3>`

  This is the desired resolution bandwidth value. Units are represented in hertz.

Example 1

```
MATH1:SPECTral:RESBw 250E3
```

This command sets the resolution bandwidth to the attainable value that is close to 250 KHz.

Example 2

```
MATH1:SPECTral:RESBw?
```

This query might return `:MATH1:SPECTRAL:RESBW 1.0002E+06`, indicating the actual resolution bandwidth value obtained from the spectral analyzer.

MATH<x>:SPECTral:SPAN

Description

This command sets the ceiling of the span to a value that is closest to the specified value. The query form of this command returns the current span value of the specified math waveform. This command is equal to selecting Spectral Setup from the Math menu and then entering a value in the Freq Span box.

Group

Math

Related Commands

MATH<x>:SPECTral:CENTER (see page 209)

Syntax 1

```
MATH<x>:SPECTral:SPAN {<NR3>|FULL}
```

Syntax 2

```
MATH<x>:SPECTral:SPAN?
```

Arguments

- `<NR3>`

  This specifies the frequency span of the output data vector from the spectral analyzer.

- `FULL`

  This sets the top of the span to 1/2 the sample rate and sets the Center frequency to 1/2 the span.

Example 1

```
MATH1:SPECTral:SPAN FULL
```

This command sets the top of the span to 1/2 the sample rate and sets the Center frequency to 1/2 the span.

Example 2

```
MATH1:SPECTral:SPAN 2.56E6
```

This command sets the top of the span to a value that is closest to 2.56 MHz.

Example 3

```
MATH1:SPECTral:SPAN?
```

This query might return `:MATH1:SPECTRAL:SPAN 1.2500E+09`, indicating the actual span value obtained by the spectral analyzer.

MATH<x>:SPECTral:SUPPress

Description

This command sets or returns the phase suppression threshold. 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.

Group

Math

Related Commands

MATH<x>:SPECTral:UNWRap (see page 220)

Syntax 1

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

Syntax 2

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

Argument

- `<NR3>`

  This is the level under which all data with magnitudes are displayed as zero phase.

Example 1

```
MATH1:SPECTral:SUPPress -62
```

This command specifies that any magnitude values less than -62 dB will have their phase output set to zero.

Example 2

```
MATH1:SPECTral:SUPPress?
```

This query might return `:MATH1:SPECTRAL:SUPPRESS -3.5000E+01`, indicating that the phase suppression threshold is currently set to -35 dB.

MATH<x>:SPECTral:UNWRap

Description

This command specifies or returns whether or not phase unwrap of the spectral analyzer output data is enabled. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then clicking the Unwrap button.

Group

Math

Related Commands

MATH<x>:SPECTral:SUPPress (see page 219)

Syntax 1

```
MATH<x>:SPECTral:UNWRap <ON|OFF>
```

Syntax 2

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

Arguments

- `ON`

  This enables phase unwrap.

- `OFF`

  This disables phase wrap.

- `<NR1>`

  A 0 disables phase wrap; any other value enables phase wrap.

Example 1

```
MATH1:SPECTral:UNWRap ON
```

This command enables phase wrap of the spectral analyzer output data.

Example 2

```
MATH1:SPECTral:UNWRap?
```

This query might return `:MATH1:SPECTRAL:UNWRAP 0`, indicating that phase unwrap of the spectral analyzer output data is disabled.

MATH<x>:SPECTral:WINdow

Description

This command sets or returns the window function used to multiply the spectral analyzer input data. 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 then 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 oscilloscope.

Group

Math

Related Commands

MATH<x>:SPECTral:RESBw (see page 217)

Syntax 1

```
MATH<x>:SPECTral:WINdow {RECTANGULAR|HAMMING|HANNING
|KAISERBESSEL|BLACKMANHARRIS|FLATTOP2|GAUSSIAN|TEKEXPONENTIAL}
```

Syntax 2

```
MATH<x>:SPECTral:WINdow?
```

Arguments

- `RECTANGULAR`

  This type of window function is equivalent to multiplying all gate data by one.

- `HAMMING`

  This type of window function is based on cosine series.

- `HANNING`

  This type of window function is based on cosine series.

- `KAISERBESSEL`

  This type of window function is based on cosine series.

- `BLACKMANHARRIS`

  This type of window function is based on cosine series.

- `GAUSSIAN`

  This type of window function has the best localization characteristics in the joint time/frequency plane.

- `TEKEXPONENTIAL`

  This type of window has an exponential non-symmetrical shape in the time domain and a triangular shape in the frequency domain.

- `FLATTOP2`

  This type of window function is a cosine series window with a flattened frequency response lobe.

Example 1

```
MATH2:SPECTral:WINdow HANNING
```

This command specifies to apply a Hanning window to the spectral analyzer input data.

Example 2

```
MATH2:SPECTral:WINdow?
```

This query might return `:MATH2:SPECTRAL:WINDOW TEKEXPONENTIAL`, indicating the window function used to multiply the spectral analyzer input data.

Measurement Command Group

Measurement Overview

You 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, you use the `VALue?` query to obtain measurement results.

Measurement commands can set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement. Other parameters, such as reference levels, have only one value, which applies to all measurements.

Measurement Commands

Command	Description
MEASUrement?	Returns all measurement parameters
MEASUrement:GATING?	Returns the measurement gating setting
MEASUrement:GATING	Sets measurement gating
MEASUrement:IMMed?	Returns all immediate measurement setup parameters
MEASUrement:IMMed:DElay?	Returns information about the immediate delay measurement
MEASUrement:IMMed:DElay:DIREction?	Returns the search direction used for immediate delay measurements
MEASUrement:IMMed:DElay:DIREction	Sets the search direction to use for immediate delay measurements
MEASUrement:IMMed:DElay:EDGE[1]?	Returns the slope of the edge used for immediate delay "from" waveform measurements
MEASUrement:IMMed:DElay:EDGE[1]	Sets the slope of the edge used for immediate delay "from" waveform measurements
MEASUrement:IMMed:DElay:EDGE2?	Returns the slope of the edge used for immediate delay "to" waveform measurements
MEASUrement:IMMed:DElay:EDGE2	Sets the slope of the edge used for immediate delay "to" waveform measurements
MEASUrement:IMMed:SOURCE[1]?	Returns the "from" source for all single channel immediate measurements
MEASUrement:IMMed:SOURCE[1]	Sets the "from" source for all single channel immediate measurements
MEASUrement:IMMed:SOURCE2?	Returns the "to" source for all single channel immediate measurements
MEASUrement:IMMed:SOURCE2	Sets the "to" source for all single channel immediate measurements
MEASUrement:IMMed:TYPe?	Returns the type of the immediate measurement
MEASUrement:IMMed:TYPe	Sets 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<n>:COUNt?	Returns the number of values accumulated since the last

	statistical reset
MEASUrement:MEAS<n>:MAXimum?	Returns the maximum value found since the last statistical reset
MEASUrement:MEAS<n>:MEAN?	Returns the mean value accumulated since the last statistical reset
MEASUrement:MEAS<n>:MINImum?	Returns the minimum value found since the last statistical reset
MEASUrement:MEAS<n>:STDdev?	Returns the standard deviation of values accumulated since the last statistical reset
MEASUrement:MEAS<x>?	Returns all measurement parameters for the displayed measurement
MEASUrement:MEAS<x>:DELay?	Returns the delay measurement parameters for specified measurement
MEASUrement:MEAS<x>:DELay:DIREction?	Returns the search direction used for delay measurements
MEASUrement:MEAS<x>:DELay:DIREction	Sets the search direction to use for delay measurements
MEASUrement:MEAS<x>:DELay:EDGE[1]?	Returns the slope of the edge used for delay "from" waveform measurements
MEASUrement:MEAS<x>:DELay:EDGE[1]	Sets the slope of the edge to use for delay "from" waveform measurements
MEASUrement:MEAS<x>:DELay:EDGE2?	Returns the slope of the edge used for "delay "to" waveform measurements
MEASUrement:MEAS<x>:DELay:EDGE2	Sets the slope of the edge to use for delay "to" waveform measurements
MEASUrement:MEAS<x>:SOURCE[1]?	Returns the channel from which measurements are taken
MEASUrement:MEAS<x>:SOURCE[1]	Sets the channel from which measurements are taken
MEASUrement:MEAS<x>:SOURCE[2]?	Returns the channel to which measurements are sent
MEASUrement:MEAS<x>:SOURCE[2]	Sets the channel to which measurements are sent
MEASUrement:MEAS<x>:STATE?	Returns whether the specified measurement slot is computed and displayed
MEASUrement:MEAS<x>:STATE	Sets whether the specified measurement slot is computed and displayed
MEASUrement:MEAS<x>:TYPe?	Returns the measurement<x> type
MEASUrement:MEAS<x>:TYPe	Sets the measurement<x> type

MEASUrement:MEAS<x>:UNIts?	Returns measurement<x> units
MEASUrement:MEAS<x>:VALue?	Returns the value of measurement<x>
MEASUrement:METHod?	Returns the method used for calculating reference levels
MEASUrement:METHod	Sets the method used for calculating reference levels
MEASUrement:REFLevel?	Returns reference level parameters
MEASUrement:REFLevel:ABSolute:HIGH?	Returns the top reference level for rise time
MEASUrement:REFLevel:ABSolute:HIGH	Sets the top reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW?	Returns the low reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW	Sets the low reference level for rise time
MEASUrement:REFLevel:ABSolute:MID?	Returns the mid reference level for measurements
MEASUrement:REFLevel:ABSolute:MID	Sets the mid reference level for measurements
MEASUrement:REFLevel:ABSolute:MID2?	Returns the mid reference level for delay "to" measurements
MEASUrement:REFLevel:ABSolute:MID2	Sets the mid reference level for delay "to" measurements
MEASUrement:REFLevel:METhod?	Returns the method for assigning high and low reference levels
MEASUrement:REFLevel:METhod	Sets the method for assigning high and low reference levels
MEASUrement:REFLevel:PERCent:HIGH?	Returns the top reference percent level for rise time
MEASUrement:REFLevel:PERCent:HIGH	Sets the top reference percent level for rise time
MEASUrement:REFLevel:PERCent:LOW?	Returns the low reference percent level for rise time
MEASUrement:REFLevel:PERCent:LOW	Sets the low reference percent level for rise time
MEASUrement:REFLevel:PERCent:MID?	Returns the mid reference percent level for waveform measurements
MEASUrement:REFLevel:PERCent:MID	Sets the mid reference percent level for waveform measurements
MEASUrement:REFLevel:PERCent:MID2?	Returns the mid reference percent level for second waveform measurements
MEASUrement:REFLevel:PERCent:MID2	Sets the mid reference percent level for second waveform measurements
MEASUrement:STATIstics:COUNt	Clears existing measurement

	statistics from memory
MEASUrement:STATIstics:MODE?	Returns which pair of statistical accumulations is displayed
MEASUrement:STATIstics:MODe	Turns management statistics on or off and sets which pair of statistical accumulations is displayed
MEASUrement:STATIstics:WEIghting?	Returns the 'time constant' for mean and standard deviation statistical accumulations
MEASUrement:STATIstics:WEIghting	Sets the 'time constant' for mean and standard deviation statistical accumulations

MEASUrement?

Description

This query-only command returns all measurement parameters in the following order: MEAS1, MEAS2, MEAS3, MEAS4, MEAS5, MEAS6, MEAS7, MEAS8, IMMED.

Group

Measurement

Syntax

```
MEASUrement?
```

Example

```
MEASUrement?
```

This query might return :MEASUREMENT:GATING OFF

```
;METHOD HISTOGRAM;IMMED:TYPE UNDEFINED;UNITS "V"
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:STATE 0
;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS2:STATE 0;TYPE UNDEFINED;UNITS "V"
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS3:STATE 0
;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;DELAY
:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS4:STATE 0;TYPE UNDEFINED;UNITS "V"
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS5:STATE 0
;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS6:STATE 0;TYPE UNDEFINED;UNITS "V"
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS7:STATE 0
;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;DELAY
:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS8:STATE 0;TYPE UNDEFINED;UNITS "V"
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:REFLEVEL:METHOD PERCENT
;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;
:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000
;MID1 50.0000;MID2 50.0000;:MEASUREMENT:STATISTICS:MODE
OFF;WEIGHTING 32
```

MEASUrement:GATING

Description

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 1

```
MEASUrement:GATING {ON|OFF|<NRf>|ZOOM|CURSOR}
```

Syntax 2

```
MEASUrement:GATING?
```

Arguments

- `ON`

  This turns on measurement gating.

- `OFF`

  This turns off measurement gating.

- `<NRf>`

  A 0 turns off measurement gating; any other value turns on measurement gating.

- `ZOOM`

  This turns on gating, using the left and right edges of the zoom box.

- `CURSOR`

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

Example 1

```
MEASUrement:GATING ON
```

This command turns on measurement gating.

Example 2

```
MEASUrement:GATING?
```

This query might return `:MEASUREMENT:GATING CURSOR`, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUrement:IMMed?

Description

This query-only command returns all immediate measurement setup parameters.

Group

Measurement

Syntax

```
MEASUrement:IMMed?
```

Example

```
MEASUrement:IMMed?
```

This query might return `:MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS`

MEASUrement:IMMed:DELay?

Description

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?
```

Example

```
MEASUrement:IMMed:DELay?
```

This query might return `:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS`

MEASUrement:IMMed:DELay:DIREction

Description

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:SOURCE2 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

Related Commands

MEASUrement:IMMed:SOURCE2 (see page 235)

Syntax 1

```
MEASUrement:IMMed:DELay:DIREction {BACKWards|FORWards}
```

Syntax 2

```
MEASUrement:IMMed:DELay:DIREction?
```

Arguments

- `BACKWards`

  This starts the search at the end of the waveform and looks for the last rising or falling edge in the waveform.

- `FORWards`

  This starts the search at the beginning of the waveform and looks for the first rising or falling edge in the waveform.

Example 1

```
MEASUrement:IMMed:DELay:DIREction FORWards
```

This command starts searching from the beginning of the waveform record and looks for the first rising or falling edge.

Example 2

```
MEASUrement:IMMed:DELay:DIREction?
```

This query 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]

Description

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:SOURCE1 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.

Group

Measurement

Related Commands

MEASUrement:IMMed:SOURCE1 (see page 234)

Syntax 1

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

Syntax 2

```
MEASUrement:IMMed:DELay:EDGE[1]?
```

Arguments

- `FALL`

  This specifies the falling edge.

- `RISe`

  This specifies the rising edge.

Example 1

```
MEASUrement:IMMed:DELay:EDGE[1] RISe
```

This command specifies that the rising edge be used for the immediate delay measurement .

Example 2

```
MEASUrement:IMMed:DELay:EDGE[1]?
```

This query 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

Description

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:SOURCE2 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

Related Commands

MEASUrement:IMMed:SOURCE2 (see page 235)

Syntax 1

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

Syntax 2

```
MEASUrement:IMMed:DELay:EDGE2?
```

Arguments

- `FALL`

  This specifies the falling edge.

- `RISe`

  This specifies the rising edge.

Example 1

```
MEASUrement:IMMed:DELay:EDGE2 RISe
```

This command specifies that the rising edge be used for the immediate delay measurement .

Example 2

```
MEASUrement:IMMed:DELay:EDGE2?
```

This query 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:SOURCE[1]

Description

This command sets or queries the source for all single channel immediate measurements and specifies the source to measure "from" when taking an immediate delay measurement or phase measurement. 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.

Group

Measurement

Related Commands

MEASUrement:IMMed:SOURCE2 (see page 235)

Syntax 1

```
MEASUrement:IMMed:SOURCE[1] {CH<x>|MATH<y>|REF<x>|HIStogram}
```

Syntax 2

```
MEASUrement:IMMed:SOURCE[1]?
```

Arguments

- `CH<x>`

  This is an input channel. The valid channel waveform range is from 1 through 4.

- `MATH<y>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<X>`

  This is a reference waveform. The valid reference waveform range is from 1 through 4.

- `HIStogram`

  This indicates histogram as the object to be measured.

Example 1

```
MEASUrement:IMMed:SOURCE[1] MATH1
```

This command specifies Math1 as the immediate measurement source .

Example 2

```
MEASUrement:IMMed:SOURCE{1}?
```

This query might return `:MEASUREMENT:IMMED:SOURCE1 CH3`, indicating that channel 3 is the immediate measurement source.

MEASUrement:IMMed:SOURCE2

Description

This command sets or queries the source for all single channel immediate measurements and specifies the source to measure "to" when taking an immediate delay measurement or phase measurement. 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 Source2 (To) setting.

Group

Measurement

Related Commands

MEASUrement:IMMed:SOURCE[1] (see page 234)

Syntax 1

```
MEASUrement:IMMed:SOURCE2 {CH<x>|MATH<y>|REF<x>}
```

Syntax 2

```
MEASUrement:IMMed:SOURCE2?
```

Arguments

- `CH<x>`

  This is an input channel. The valid channel waveform range is from 1 through 4.

- `MATH<y>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<X>`

  This is a reference waveform. The valid reference waveform range is from 1 through 4.

Example 1

```
MEASUrement:IMMed:SOURCE2 REF3
```

This command sets the waveform in reference memory location 3 as the delay "to" source when making delay measurements.

Example 2

```
MEASUrement:IMMed:SOURCE2?
```

This query might return `:MEASUREMENT:IMMED:SOURCE2 MATH1`, indicating that Math1 is the immediate measurement source.

MEASUrement:IMMed:TYPe

Description

This command sets or queries the immediate measurement type.

Group

Measurement

Syntax 1

```
MEASUrement:IMMed:TYPE {AMPlitude|AREa|BURst|CARea|CMEam|
CRMs|DELay|FALL|FREQuency|HIGH|LOW|MAXimum|MEAN|MINImum|
NDUty|NOVershoot|NWIdth|PDUty|PERIod|PHAse|PK2Pk|POVershoot|
PWIdth|RISe|RMS}
```

Syntax 2

```
MEASUrement:IMMed:TYPe?
```

Arguments

- `AMPlitude`

  This is the high value minus the low value.

- `AREa`

  This is the area between the curve and ground over the entire waveform.

- `BURst`

  This is the time from the first MidRef crossing to the last MidRef crossing.

- `CARea`

  This is the area between the curve and ground over one cycle.

- `CMEan`

  This is the arithmetic mean over one cycle.

- `CRMs`

  This is the true Root Mean Square voltage over one cycle.

- `DELay`

  This is the time between the MidRef crossings of two different waveforms.

- `FALL`

  This is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value of its final value.

- `FREQuency`

  This is the reciprocal of the period measured in hertz.

- `HIGH`

  This is the 100% reference level.

- `LOW`

  This is the 0% reference level.

- `MAXimum`

  This is the highest amplitude (voltage).

- `MINImum`

  This is the lowest amplitude (voltage).

- `NDUty`

  This is the ratio of the negative pulse width to the signal period expressed as a percentage.

- `NOVershoot`

This is the negative overshoot, expressed as 100 x (Low - Minimum/Amplitude).

- `NWIdth`

  This is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

- `PDUty`

  This is the ratio of the positive pulse width to the signal period, expressed as a percentage.

- `PERIod`

  This is the time, in seconds, it takes for one complete signal cycle to happen.

- `PHAse`

  This is the amount of shift, expressed in degrees, from the selected waveform to the designated waveform.

- `PK2Pk`

  This is the absolute difference between the maximum and minimum amplitude.

- `POVershoot`

  This is the positive overshoot, expressed as 100 x (Maximum-High/Amplitude).

- `PWIdth`

  This is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

- `RISe`

  This is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value of its final value.

- `RMS`

  This is the true Root Mean Square voltage.

Example 1

```
MEASUrement:IMMed:TYPe FREQuency
```

This defines the immediate measurement to be a frequency measurement.

Example 2

```
MEASUrement:IMMed:TYPe?
```

This query might return `:MEASUREMENT:IMMED:TYPe RMS`, that the immediate measurement is the true Root Mean Square voltage.

MEASUrement:IMMed:UNIts?

Description

This query-only command returns the units of the immediate measurement.

Group

Measurement

Syntax

```
MEASUrement:IMMed:UNIts?
```

Example

```
MEASUrement:IMMed:UNIts?
```

This query might return `:MEASUREMENT:IMMED:UNIts "s"`, indicating that units for the immediate measurement are represented in seconds.

MEASUrement:IMMed:VALue?

Description

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 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

Related Commands

MEASUrement:IMMed:TYPe (see page 236), MEASUrement:IMMed:SOURCE[1] (see page 234), MEASUrement:IMMed:SOURCE2 (see page 235), *ESR? (see page 320), ALLEV? (see page 329)

Syntax

```
MEASUrement:IMMed:VALue?
```

Example

```
MEASUrement:IMMed:VALue?
```

This query 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<n>:COUNt?

Description

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.

Group

Measurement

Syntax

```
MEASUrement:MEAS<n>:COUNt?
```

Example

```
MEASUrement:MEAS3:COUNt?
```

This query might return `:MEASUREMENT:MEAS3:COUNT 3.247000000E+03`

MEASUrement:MEAS<n>:MAXimum?

Description

This query-only command returns the maximum value found for this measurement since the last statistical reset.

Group

Measurement

Syntax

```
MEASUrement:MEAS<n>:MAXimum?
```

Example

```
MEASUrement:MEAS3:MAXimum?
```

This query might return `:MEASUREMENT:MEAS3:MAXIMUM 4.18E-9`

MEASUrement:MEAS<n>:MEAN?

Description

This query-only command returns the mean value accumulated for this measurement since the last statistical reset.

Group

Measurement

Syntax

```
MEASUrement:MEAS<n>:MEAN?
```

Example

```
MEASUrement:MEAS1:MEAN?
```

This query might return `:MEASUREMENT:MEAS1:MEAN 514.71E-09`

MEASUrement:MEAS<n>:MINImum?

Description

This query-only command returns the minimum value found for this measurement since the last statistical reset.

Group

Measurement

Syntax

```
MEASUrement:MEAS<n>:MINImum?
```

Example

```
MEASUrement:MEAS1:MINImum?
```

This query might return `:MEASUREMENT:MEAS1:MINIMUM 1.75E-09`

MEASUrement:MEAS<n>:STDdev?

Description

This query-only command returns the standard deviation of values accumulated for this measurement since the last statistical reset.

Group

Measurement

Syntax

```
MEASUrement:MEAS<n>:STDdev?
```

Example

```
MEASUrement:MEAS1:STDdev?
```

This query might return `:MEASUREMENT:MEAS1:STDDEV 21.0E-12`

MEASUrement:MEAS<x>?

Description

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>?
```

Example

```
MEASUrement:MEAS3?
```

This query might return `:MEASUREMENT:MEAS3:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS`

MEASUrement:MEAS<x>:DELay?

Description

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 then viewing the Delay Edge and Search Direction settings.

Group

Measurement

Syntax 1

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

Example

```
MEASUrement:MEAS3?
```

This query might return `:MEASUREMENT:MEAS3:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS`

MEASUrement:MEAS<x>:DELay:DIREction

Description

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>: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 Search Direction setting.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:SOURCE2 (see page 251), MEASUrement:MEAS<x>:DELay:EDGE2 (see page 249)

Syntax 1

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

Syntax 2

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

Arguments

- `BACKWards`

  This 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:EDGE2 command to specify the slope of the edge.

- `FORWards`

  This 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:EDGE2 command to specify the slope of the edge.

Example 1

```
MEASUrement:MEAS3:DELay:DIREction BACKWards
```

This command starts searching from the end of the waveform record.

Example 2

```
MEASUrement:MEAS3DELay:DIREction?
```

This query might return `:MEASUREMENT:MEAS3:DELAY:DIRECTION BACKWARDS` indicating the current search direction.

MEASUrement:MEAS<x>:DELay:EDGE[1]

Description

This command sets or queries the slope of the edge that is used for the delay "from" waveform when taking a delay measurement. Use the MEASUrement:MEAS<x>:SOURCE[1] 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 Edge1 setting.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:SOURCE[1] (see page 250)

Syntax 1

```
MEASUrement:MEAS<x>:DELay:EDGE[1] {FALL|RISe}
```

Syntax 2

```
MEASUrement:MEAS<x>:DELay:EDGE[1]?
```

Arguments

- `FALL`

  This specifies the falling edge of the waveform.

- `RISe`

  This specifies the rising edge of the waveform.

Example 1

```
MEASUrement:MEAS3:DELay:EDGE[1] RISe
```

This command specifies that the rising edge of the waveform be used for measurement 3.

Example 2

```
MEASUrement:MEAS1:DELay:EDGE[1]?
```

This query might return `:MEASUREMENT:MEAS1:DELAY:EDGE[1] FALL`, indicating that the falling edge of the waveform is being used for measurement 1.

MEASUrement:MEAS<x>:DELay:EDGE2

Description

This command sets or queries the slope of the edge that is used for the delay "to" waveform 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 Edge2 setting.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:SOURCE2 (see page 251)

Syntax 1

```
MEASUrement:MEAS<x>:DELay:EDGE2 {FALL|RISe}
```

Syntax 2

```
MEASUrement:MEAS<x>:DELay:EDGE2?
```

Arguments

- `FALL`

  This specifies the falling edge of the waveform.

- `RISe`

  This specifies the rising edge of the waveform.

Example 1

```
MEASUrement:MEAS3:DELay:EDGE2 RISe
```

This command specifies that the rising edge of the waveform be used for measurement 3.

Example 2

```
MEASUrement:MEAS1:DELay:EDGE2?
```

This query might return `:MEASUREMENT:MEAS1:DELAY:EDGE2 FALL`, indicating that the falling edge of the waveform is being used for measurement 1.

MEASUrement:MEAS<x>:SOURCE[1]

Description

This command sets or queries the source for all single channel measurements and specifies the source to measure "from" 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 and then choosing the desired measurement source.

Group

Measurement

Syntax 1

```
MEASUrement:MEAS<x>:SOURCE[1] {CH<x>|MATH<y>|REF<x>|HIStogram}
```

Syntax 2

```
MEASUrement:MEAS<x>:SOURCE[1]?
```

Arguments

- `CH<x>`

  This is an input channel. The valid channel waveform range is from 1 through 4.

- `MATH<y>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This is a reference waveform. The valid reference waveform range is from 1 through 4.

- `HIStogram`

  This is a histogram.

Example 1

```
MEASUrement:MEAS2:SOURCE[1] MATH1
```

This command specifies Math 1 as the measurement 2 source.

Example 2

```
MEASUrement:MEAS1:SOURCE[1]?
```

This query might return `:MEASUREMENT:MEAS1:SOURCE[1] MATH1`, indicating that Math1 is the measurement 2 source.

MEASUrement:MEAS<x>:SOURCE2

Description

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

Related Commands

MEASU:MEAS<x>:TYPe (see page 253)

Syntax 1

```
MEASUrement:MEAS<x>:SOURCE2 {CH<x>|MATH<y>|REF<x>}
```

Syntax 2

```
MEASUrement:MEAS<x>:SOURCE2?
```

Arguments

- `CH<x>`

  This is an input channel. The valid channel waveform range is from 1 through 4.

- `MATH<y>`

  This is a math waveform. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This is a reference waveform . The valid reference waveform range is from 1 through 4.

Example 1

```
MEASUrement:MEAS4:SOURCE2 CH1
```

This command specifies CH1 as the delay "to" source when making delay measurement.

Example 2

```
MEASUrement:MEAS2:SOURCE2?
```

This query might return `:MEASUREMENT:MEAS2:SOURCE2 MATH1`, indicating that Math 1 is the measurement 2 source.

MEASUrement:MEAS<x>:STATE

Description

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[1] command. You define the measurement type using the MEASUrement:MEAS<x>:TYPe command.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:SOURCE[1] (see page 250), MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax 1

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

Syntax 2

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

Arguments

- `OFF`

  This disables calculation and display of the specified measurement slot.

- `ON`

  This enables calculation and display of the specified measurement slot.

- `<NR1>`

  A 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.

Example 1

```
MEASUrement:MEAS2:STATE ON
```

This command computes and displays the measurement defined as measurement 2.

Example 2

```
MEASUrement:MEAS1:STATE?
```

This query might return `:MEASUREMENT:MEAS1:STATE 0`, indicating that measurement defined for measurement slot 1 is disabled.

MEASUrement:MEAS<x>:TYPe

Description

This command sets or queries the type of measurement that is defined for the specified measurement slot. The measurement slot is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement type.

Group

Measurement

Syntax 1

```
MEASUrement:MEAS<x>:TYPe {AMPlitude|AREa|BURst|CARea|CMEam|CRMs|DELay|FALL
|FREQuency|HIGH||LOW|MAXimum|MEAN|MINImum|NDUty|NOVershoot|NwIdth|PDUty
|PERIod|PHAse|PK2Pk|POVershoot|PWIdth|RISe|RMS|UNDEFINED}
```

Syntax 2

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

Arguments

- `AMPlitude`

  This measures the amplitude of the selected waveform.

- `AREa`

  This 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`

  This measures the duration of a burst. The measurement is made over the entire waveform or gated region.

- `CARea`

  This measures the voltage over time. The area is over the first cycle in the waveform or the first cycle in the gated region and measured in volt-seconds. The area measured above ground is positive, while the area below ground is negative.

- `CMEam`

  This measures the arithmetic mean over the first cycle in the4 waveform or the first cycle in the gated region..

- `CRMs`

  This measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

- `DELay`

  This measures the time between the middle reference point of the source waveform and the destination waveform.

- `FALL`

  This 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`

  This 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`

  This 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.

- `LOW`

  This measures the Low reference (0% level, sometimes called Baseline) of a waveform.

- `MAXimum`

  This finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region.

- `MEAN`

  This amplitude measurement finds the arithmetic mean over the entire waveform or gated region.

- `MINImum`

  This amplitude measurement finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region.

- `NDUty`

  This measures the first cycle in the waveform or gated region. The negative duty cycle is the ratio of the negative pulse width to the signal period, expressed as a percentage.

- `NOVershoot`

  This amplitude measurement finds the negative overshoot value over the entire waveform or gated region.

- `NWIdth`

  This measures the first pulse in the waveform or gated region. This distance (time) is measured at the middle reference amplitude points of a negative pulse.

- `PDUty`

  This measures the first cycle in the waveform or gated region. The positive duty cycle is the ratio of the positive pulse width to the signal period, expressed as a percentage.

- `PERIod`

  This timing measurement finds the time it takes for the first complete waveform cycle to complete in the waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

- `PHAse`

  This 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`

  This peak-to-peak amplitude measurement finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region.

- `POVershoot`

  This amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

- `PWIdth`

  This measures the first pulse in the waveform or gated region. The distance (time) is measured at the middle reference amplitude points of a positive pulse.

- `RISe`

  This 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%).

  Rise time = TimeRef High - TimeRef Low

  You can also limit the measurement (normally taken over the entire waveform record) to a gated region on the waveform.

- `RMS`

This amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

- `UNDEFINED`

  This is the default measurement type, which indicates that a measurement type has not yet been chosen. Once a measurement type is chosen, measurements cannot be reset using this argument.

Example 1

```
MEASUrement:MEAS2:TYPe FREQuency
```

This command defines measurement 2 as a measurement of the frequency of a waveform.

Example 2

```
MEASUrement:MEAS1:TYPe?
```

This query might return `:MEASUREMENT:MEAS1:TYPE RMS`, indicating that measurement 1 is defined to measure the RMS value of a waveform.

MEASUrement:MEAS<x>:UNIts?

Description

This query-only command returns the units associated with the specified measurement.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax

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

Example

```
MEASUrement:MEAS1:UNIts?
```

This query might return `:MEASUREMENT:MEAS1:UNIts %`, indicating units for measurement 1 are set to percent.

MEASUrement:MEAS<x>:VALue?

Description

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 onscreen 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 oscilloscope may take longer to update.

Group

Measurement

Related Commands

MEASUrement:MEAS<x>:UNIts? (see page 256), *ESR? (see page 320), ALLEV? (see page 329)

Syntax

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

Example

```
MEASUrement:MEAS1:VALue?
```

This query 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

Description

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 Form setting.

Group

Measurement

Related Commands

MEASUrement:REFLevel:PERCent:HIGH (see page 265), MEASUrement:REFLevel:PERCent:LOW (see page 266), MEASUrement:REFLevel:PERCent:MID (see page 267), MEASUrement:REFLevel:PERCent:MID2 (see page 268)

Syntax 1

```
MEASUrement:METHod {HIStogram|MINMax)
```

Syntax 2

```
MEASUrement:METHod?
```

Arguments

- `HIStogram`

  This sets the high and low reference levels statistically using a histogram algorithm.

- `MINMax`

  This sets the high and low waveform levels to MAX and MIN, respectively.

Example 1

```
MEASUrement:METHod HIStogram
```

This command specifies that the high and low reference levels are set statistically.

Example 2

```
MEASUrement:METHod?
```

This query might return `:MEASUREMENT:METHOD MINMAX`, indicating that the reference levels are set to MIN and MAX.

MEASUrement:REFLevel?

Description

This query-only command returns the current reference level parameters. This command is equivalent to selecting Reference Levels from the Measure menu and then viewing the current Reference Levels settings.

Group

Measurement

Syntax

```
MEASUrement:REFLevel?
```

Example

```
MEASUrement:REFLevel?
```

This query might return `:MEASUREMENT:REFLEVEL:METHOD ABSOLUTE ;ABSOLUTE:HIGH 0.0000E+00;LOW 0.0000E+00;MID1 0.0000E+00 ;MID2 0.0000E+00;:MEASUREMENT:REFLEVEL:PERCENT:HIGH 9.0000E+01;LOW 1.0000+01;MID1 5.0000E+01;MID2 5.0000E+01`

MEASUrement:REFLevel:ABSolute:HIGH

Description

This command sets or queries the high reference level, and is the upper reference level when MEASUrement:REFLevel:METHod is set to Absolute. 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 Absolute High Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264), MEASUrement:IMMed:TYPe (see page 236), MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax 1

```
MEASUrement:REFLevel:ABSolute:HIGH <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:ABSolute:HIGH?
```

Argument

- `<NR3>`

  This is the high reference level, in volts. The default is 0.0 V.

Example 1

```
MEASUrement:REFLevel:ABSolute:HIGH 1.71
```

This command sets the high reference level to 1.71 V.

Example 2

```
MEASUrement:REFLevel:ABSolute:HIGH?
```

This query 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

Description

This command sets or queries the low reference level, and is the lower reference level when MEASUrement:REFLevel:METHod is set to Absolute. 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 Absolute Low Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264), MEASUrement:IMMed:TYPe (see page 236), MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax 1

```
MEASUrement:REFLevel:ABSolute:LOW <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:ABSolute:LOW?
```

Argument

- `<NR3>`

  This is the low reference level, in volts. The default is 0.0 V.

Example 1

```
MEASUrement:REFLevel:ABSolute:LOW 0.0 V.
```

This command sets the low reference level to 0.0 V.

Example 2

```
MEASUrement:REFLevel:ABSolute:LOW?
```

This query 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

Description

This command sets or queries the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to Absolute. Note that this command affects the results of period, frequency, delay, and all cyclic measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Absolute Mid Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264)

Syntax 1

```
MEASUrement:REFLevel:ABSolute:MID[1] <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:ABSolute:MID[1]?
```

Argument

- `<NR3>`

  This is the mid reference level, in volts. The default is 0.0 V.

Example 1

```
MEASUrement:REFLevel:ABSolute:MID[1] .71
```

This command sets the mid reference level to .71 V.

Example 2

```
MEASUrement:REFLevel:ABSolute:MID[1]?
```

This query might return `:MEASUREMENT:REFLEVEL:ABSOLUTE:MID 0.7100E+00`, indicating that the absolute mid1 reference level is set to .71 V.

MEASUrement:REFLevel:ABSolute:MID2

Description

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:REFLevel:METHod is set to Absolute. Note that this command affects the results of delay measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Absolute Mid2 Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264)

Syntax 1

```
MEASUrement:REFLevel:ABSolute:MID2 <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:ABSolute:MID2?
```

Argument

- `<NR3>`

  This is the mid reference level, in volts. The default is 0.0 V.

Example 1

```
MEASUrement:REFLevel:ABSolute:MID2 0.5
```

This command sets the mid reference level for the delay waveform to 0.5 V.

Example 2

```
MEASUrement:REFLevel:ABSolute:MID2?
```

This query 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

Description

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 1

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

Syntax 2

```
MEASUrement:REFLevel:METHod?
```

Arguments

- `ABSolute`

  This specifies that the reference levels are set explicitly using the MEASUrement: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`

  This 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.

Example 1

```
MEASUrement:REFLevel:METHod ABSolute
```

This command specifies that explicit user-defined values are used for the reference levels.

Example 2

```
MEASUrement:REFLevel:METHod?
```

This query 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

Description

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. 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 High Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264), MEASUrement:IMMed:TYPe (see page 236), MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax 1

```
MEASUrement:REFLevel:PERCent:HIGH <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:PERCent:HIGH?
```

Argument

- `<NR3>`

  This is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

Example 1

```
MEASUrement:REFLevel:PERCent:HIGH 95
```

This command sets the high reference level to 95% of HIGH.

Example 2

```
MEASUrement:REFLevel:PERCent:HIGH?
```

This query might return `:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90`, indicating that the percentage high reference level is set to 90% of HIGH.

MEASUrement:REFLevel:PERCent:LOW

Description

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. 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.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264), MEASUrement:IMMed:TYPe (see page 236), MEASUrement:MEAS<x>:TYPe (see page 253)

Syntax 1

```
MEASUrement:REFLevel:PERCent:LOW <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:PERCent:LOW?
```

Argument

- `<NR3>`

  This is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

Example 1

```
MEASUrement:REFLevel:PERCent:LOW 15
```

This command sets the high reference level to 15% of HIGH.

Example 2

```
MEASUrement:REFLevel:PERCent:LOW?
```

This query might return `:MEASUREMENT:REFLEVEL:PERCENT:LOW 10`, indicating that the percentage high reference level is set to 10% of HIGH.

MEASUrement:REFLevel:PERCent:MID[1]

Description

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. Note that this command affects the results of period, frequency, delay, and all cyclic measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Percentage Mid Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264)

Syntax 1

```
MEASUrement:REFLevel:PERCent:MID[1] <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:PERCent:MID[1]?
```

Argument

- `<NR3>`

  This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

Example 1

```
MEASUrement:REFLevel:PERCent:MID[1] 60
```

This command sets the mid reference level to 60% of HIGH.

Example 2

```
MEASUrement:REFLevel:PERCent:MID[1]?
```

This query might return `:MEASUREMENT:REFLEVEL:PERCENT:MID 65`, indicating that the percentage mid reference level is set to 65% of HIGH.

MEASUrement:REFLevel:PERCent:MID2

Description

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when MEASUrement:REFLevel:METHod is set to Percent. Note that this command affects the results of delay measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Percentage Mid2 Ref value.

Group

Measurement

Related Commands

MEASUrement:REFLevel:METHod (see page 264)

Syntax 1

```
MEASUrement:REFLevel:PERCent:MID2 <NR3>
```

Syntax 2

```
MEASUrement:REFLevel:PERCent:MID2?
```

Argument

- `<NR3>`

  This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

Example 1

```
MEASUrement:REFLevel:PERCent:MID2 40
```

This command sets the mid2 reference level to 40% of HIGH.

Example 2

```
MEASUrement:REFLevel:PERCent:MID2?
```

This query might return `:MEASUREMENT:REFLEVEL:PERCENT:MID2 45`, indicating that the percentage mid2 reference level is set to 45% of HIGH.

MEASUrement:STATIstics:COUNt

Description

This command (no query form) clears existing measurement statistics from memory. This command is equivalent to selecting Reset Statistics from the Measure menu.

Group

Measurement

Related Commands

MEASUrement:STATIstics:MODe (see page 270)

Syntax

```
MEASUrement:STATIstics:COUNt {RESET}
```

Argument

- `RESET`

  This clears existing measurement statistics from memory.

Example

```
MEASUrement:STATIstics:COUNt RESET
```

This command clears existing measurement statistics from memory.

MEASUrement:STATIstics:MODe

Description

This command controls the operation and display of management 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

Related Commands

MEASUrement:STATIstics:WEIghting (see page 271)

Syntax 1

```
MEASUrement:STATIstics:MODe {OFF|ALL|VALUEMean}
```

Syntax 2

```
MEASUrement:STATIstics:MODe?
```

Arguments

- `OFF`

  This turns measurements off. This is the default value.

- `ALL`

  This turns on statistics and displays all statistics for each measurement.

- `VALUEMean`

  This turns on statistics and displays the value and the mean (μ) of each measurement.

Example 1

```
MEASUrement:STATIstics:MODe OFF
```

This command turns statistic measurements off.

Example 2

```
MEASUrement:STATIstics:MODe?
```

This query 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

Description

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 then entering the desired Weight n= value.

Group

Measurement

Related Commands

MEASUrement:STATIstics:MODE (see page 270)

Syntax 1

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

Syntax 2

```
MEASUrement:STATIstics:WEIghting?
```

Argument

- `<NR1>`

  This is the time constant for the mean and standard deviation statistical accumulations.

Example 1

```
MEASUrement:STATIstics:WEIghting 4
```

This command sets statistical weighting to 4.

Example 2

```
MEASUrement:STATIstics:WEIghting?
```

This query might return `:MEASUREMENT:STATISTICS:WEIGHTING 4`, indicating that measurement statistics weighting is currently set to 4.

Miscellaneous Command Group

Miscellaneous Overview

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.

Miscellaneous Commands

Command	Description
*DDT?	Returns the commands that are executed by the group execute trigger
*DDT	Sets the commands that will be executed by the group execute trigger
*IDN?	Returns the oscilloscope identification code
*LRN?	Returns a listing of oscilloscope settings
*TRG	Performs the group execute trigger (GET)
*TST?	Tests the GPIB interface and returns status
AUTOSet EXECute	Adjusts vertical, horizontal and trigger controls to provide a stable display of the selected waveform
AUXout?	Returns the auxiliary out setup
AUXout:EDGE?	Returns the polarity of the trigger output signal
AUXout:EDGe	Sets the polarity of the trigger output signal
AUXout:SOUrce?	Returns the trigger source at the BNC connection
AUXout:SOUrce	Sets the trigger source at the BNC connection
BELl	In previous TDS models, beeped the audio indicator. Provided for backward compatibility.
CMDBatch?	Returns the state of command batching
CMDBatch	Turns command batching on or off
DATe?	Returns the date that the oscilloscope can display
DATe	Sets the date that the oscilloscope can display
HDR?	Returns the Response Header Enable State
HDR	Sets the Response Header Enable State
HEADer?	Returns the Response Header Enable State
HEADer	Sets the Response Header Enable State
ID?	Returns identifying information about the instrument and its firmware
LOCK?	Query front panel lock state
LOCK	Sets 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 oscilloscope
SET?	Returns a listing of oscilloscope settings
TIMe?	Returns the time displayed by the oscilloscope
TIMe	Sets the time displayed by the oscilloscope
UNLock	Unlocks front panel
VERBose?	Returns verbose state
VERBose	Sets verbose state

*DDT

Description

This command allows you to specify a command or a list of commands that are executed when the oscilloscope 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

Related Commands

ALIas (see page 33), *TRG (see page 279)

Syntax 1

```
*DDT {<Block>|<Qstring>}
```

Syntax 2

```
*DDT?
```

Arguments

- `<Block>`

  This 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>`

  This 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.

Example

```
*DDT #OACQUIRE:STATE RUN
```

This command specifies that the acquisition system will be started each time a *TRG command is sent.

*IDN?

Description

This query-only command returns the oscilloscope identification code.

Group

Miscellaneous

Related Commands

ID? (see page 290)

Syntax

```
IDN?
```

Example

```
*IDN?
```

This query might return `:TEKTRONIX,TDS7104,0,CF:91.1CT FV:01.00.912`, indicating the oscilloscope model number, configured number, and firmware version number.

*LRN?

Description

This query-only command returns the commands that list the oscilloscope settings (except for configuration information for the calibration values), allowing you to record or "learn" the current oscilloscope settings. You can use these commands to return the oscilloscope to the state it was in when you made the *LRN? query.

Group

Miscellaneous

Related Commands

SET?

Syntax

```
*LRN?
```

Example

```
*LRN?
```

This query might return the following response:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV
10;NUMAVG 16;REPET 1;:FASTACQ:STATE 1;:APPLICATION
:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;
:APPLICATION:WINDOW:HEIGHT 0;WIDTH 0;:APPLICATION
:SCOPEAPP:STATE NOTRUNNING;WINDOW FULLSCREEN;
:APPLICATION:EXTAPP:STATE NOTRUNNING;:AUXOUT:SOURCE
ATRIGGER;EDGE FALLING;:CMDBATCH 1;:HEADER 0;:LOCK
NONE;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1
;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR
DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE
FULL;INTENSITY:WAVEFORM 60.0000;AUTOBRIGHT 1
;SCREENSAVER 1;SCREENSAVERDELAY 28800;:DISPLAY
:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1
;VARPERSIST 500.0000E-3;:HARDCOPY:FILENAME "";FORMAT
BMP;LAYOUT PORTRAIT;PALETTE HARDCOPY;PORT FILE;
:HORIZONTAL:DELAY:MODE 1;POSITION 50.0000;TIME
0.0000;:HORIZONTAL:MAIN:SCALE 200.0000E-9;POSITION
50.0000;:HORIZONTAL:RECORDLENGTH 2500;ROLL AUTO
;FASTFRAME:STATE 0;LENGTH 2500;COUNT 2;SELECTED:CH1 2
;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2
;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME
:REF:SOURCE CH1;FRAME 1;:HORIZONTAL:FASTFRAME:TRACK
LIVE;:DIAG:LEVEL SUBSYS;:TRIGGER:A:MODE AUTO;TYPE
EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 250.0000E-9;
:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
: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: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: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 SLOWERTHAN;:TRIGGER:A:PULSE
:WIDTH:LOWLIMIT 2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN
;POLARITYPOSITIVE;:TRIGGER:A:PULSE:TIMEOUT:POLARITY
STAYSHIGH;TIME 2.0000E-9;: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;
:MATH1:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 65;:MATH1:SPECTRAL:MAG DB
```

```
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH2:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 80;:MATH2:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH3:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 95;:MATH3:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH4:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 110;:MATH4:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3
;SPAN 600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6
;WINDOW GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0
;:HISTOGRAM:BOXPCNT 30.0000,25.1000,70.0000,75.2000
;DISPLAY LINEAR;STATE 0;FUNCTION HORIZONTAL;SIZE 2.0000
;SOURCE CH1;:CH1:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW
0.0000;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH1:LABEL:NAME "";XPOS 5;YPOS 5;
:CH2:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH2:LABEL:NAME "";XPOS 5;YPOS 20;
:CH3:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH3:LABEL:NAME "";XPOS 5;YPOS 35;
:CH4:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH4:LABEL:NAME "";XPOS 5;YPOS 50;
: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;:CURSOR:STATE OFF;FUNCTION VBARS;MODE INDEPENDENT
;SOURCE CH1;VBARS:UNITS SECONDS;POSITION1 -800.0000E-9
;POSITION2 800.0000E-9;:CURSOR:HBARS:POSITION1
300.0000E-3;POSITION2 -300.0000E-3;:CURSOR:PAIRED
:POSITION1 -800.0000E-9;POSITION2 800.0000E-9;:CURSOR
:SPLIT:POSITION1 -800.0000E-9;POSITION2 800.0000E-9
;SOURCE2 CH1;UNITS BASE;MEASUREMENT:GATING OFF;METHOD
HISTOGRAM;IMMED:TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;SOURCE1 CH1
;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION
FORWARDS;:MEASUREMENT:MEAS2:STATE 0;TYPE UNDEFINED
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS3:STATE 0
;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1
RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS4
:STATE 0;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS5:STATE 0;TYPE UNDEFINED;SOURCE1 CH1
;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION
FORWARDS;:MEASUREMENT:MEAS6:STATE 0;TYPE UNDEFINED
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS7:STATE 0;TYPE
UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE
;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS8
:STATE 0;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY
:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT
:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000
;MID1 0.0000;MID2 0.0000;:MEASUREMENT:REFLEVEL:PERCENT
```

```
:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;
:MEASUREMENT:STATISTICS:MODE OFF;WEIGHTING 32;:ZOOM
:MODE 0;GRATICULE:SIZE 50;:ZOOM:HORIZONTAL:LOCK ALL;
:ZOOM:CH1:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM
:CH1:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM:CH2
:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:CH2:VERTICAL
:POSITION 0.0000;SCALE 1.0000;:ZOOM:CH3:HORIZONTAL
:POSITION 50.0000;SCALE 2;:ZOOM:CH3:VERTICAL:POSITION
0.0000;SCALE 1.0000;:ZOOM:CH4:HORIZONTAL:POSITION
50.0000;SCALE 2;:ZOOM:CH4:VERTICAL:POSITION 0.0000
;SCALE 1.0000;:ZOOM:MATH1:HORIZONTAL:POSITION 50.0000
;SCALE 2;:ZOOM:MATH1:VERTICAL:POSITION 0.0000;SCALE
1.0000;:ZOOM:MATH2:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH2:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH3:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM
:MATH3:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM
:MATH4:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:MATH4
:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM:REF1
:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:REF1:VERTICAL
:POSITION 0.0000;SCALE 1.0000;:ZOOM:REF2:HORIZONTAL
:POSITION 50.0000;SCALE 2;:ZOOM:REF2:VERTICAL:POSITION
0.0000;SCALE 1.0000;:ZOOM:REF3:HORIZONTAL:POSITION
50.0000;SCALE 2;:ZOOM:REF3:VERTICAL:POSITION 0.0000
;SCALE 1.0000;:ZOOM:REF4:HORIZONTAL:POSITION 50.0000
;SCALE 2;:ZOOM:REF4:VERTICAL:POSITION 0.0000;SCALE
1.0000;:REF1:LABEL:NAME "";XPOS 5;YPOS 125;:REF1:SCALE
1.0000;POSITION 0.0000;:REF2:LABEL:NAME "";XPOS 5;YPOS
140;:REF2:SCALE 1.0000;POSITION 0.0000;:REF3:LABEL:NAME
"";XPOS 5;YPOS 155;:REF3:SCALE 1.0000;POSITION 0.0000;
:REF4:LABEL:NAME "";XPOS 5;YPOS 170;:REF4:SCALE 1.0000
;POSITION 0.0000;:DATA:DESTINATION REF1;ENCDG RIBINARY
;SOURCE CH1;START 1;STOP 500;WIDTH 8
```

*TRG

Description

This command (no query form) performs the group execute trigger on commands defined by *DDT.

Group

Miscellaneous

Related Commands

*DDT (see page 274)

Syntax

```
*TRG
```

Example

```
*TRG
```

This command immediately executes all commands that have been defined by *DDT.

*TST?

Description

This query-only command tests (self-test) the GPIB interface and returns a 0.

Group

Miscellaneous

Syntax

```
*TST?
```

Example

```
*TST?
```

This query always returns `0`.

AUTOSet EXECute

Description

This command causes the digitizing oscilloscope to adjust its vertical, horizontal, and trigger controls to provide a stable display of the selected waveform. This is equivalent to pressing the front-panel **AUTOSET** button. For a detailed description of autoset functionality, see *Autoset* in the index of your oscilloscope online help.

Group

Miscellaneous

Syntax

```
AUTOSet EXECute
```

Argument

- `EXECute`

  This autosets the displayed waveform.

Example

```
AUTOSet EXECute
```

This command adjusts the vertical, horizontal, and trigger controls to provide a stable display of the selected waveform.

AUXout?

Description

This query-only command returns the auxiliary out setup. This query command is equivalent to selecting AUX OUT Configuration from the Utilities menu and then viewing the current settings.

Group

Miscellaneous

Related Commands

AUXout:EDGe (see page 283), AUXout:SOUrce (see page 284)

Syntax

```
AUXout?
```

Example

```
AUXout?
```

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

AUXout:EDGE

Description

This command sets or queries the direction in which the trigger output signal transitions when a trigger occurs. This command is equivalent to selecting AUX OUT Configuration from the Utilities menu and then clicking the desired Polarity setting.

Group

Miscellaneous

Related Commands

AUXout? (see page 282), AUXout:SOUrce (see page 284)

Syntax 1

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

Syntax 2

```
AUXout:EDGE?
```

Arguments

- `RISing`

  This sets the polarity to the rising edge of the trigger output signal.

- `FALling`

  This sets the polarity to the falling edge of the trigger output signal.

Example 1

```
AUXout:EDGE RISing
```

This command sets the polarity to the rising edge of the trigger output signal.

Example 2

```
AUXout:EDGE?
```

This query might return `:AUXOUT:EDGE RISING`, indicating that the polarity is set to the rising edge of the trigger output signal.

AUXout:SOUrce

Description

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

Related Commands

AUXout? (see page 282), AUXout:EDGe (see page 283)

Syntax 1

```
AUXout:SOUrce {ATRIGger|BTRIGger}
```

Syntax 2

```
AUXout:SOUrce?
```

Arguments

- `ATRIGger`

  This sets the source at the BNC connector to the main trigger.

- `BTRIGger`

  This sets the source at the BNC connector to the delayed trigger.

Example

```
AUXout:SOUrce?
```

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

BELI

Description

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

Group

Miscellaneous

Syntax

```
BEL1
```

Example

```
BEL1
```

This command is accepted but does nothing.

CMDBatch

Description

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 1

```
CMDBatch {OFF|ON|<NR1>}
```

Syntax 2

```
CMDBatch?
```

Arguments

- `OFF`

  This turns command batching off.

- `ON`

  This turns command batching on.

- `<NR1>`

  A 0 turns command batching off; any other value turns command batching on.

Example 1

```
CMDBatch OFF
```

This command disables command batching.

Example 2

```
CMDBatch?
```

This query might return `:CMDBATCH 1`, indicating that command batching is turned on.

DATe

Description

This command sets or queries the date that the oscilloscope 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

Related Commands

TIMe (see page 298)

Syntax 1

```
DATe <Qstring>
```

Syntax 2

```
DATe?
```

Argument

- `<QString>`

  This 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.

Example 1

```
DATE "2000-01-24"
```

This command specifies that the date is set to January 24, 2000.

Example 2

```
DATE?
```

This query might return `:DATE 2000-01-24`, indicating the current date is set to January 24, 2000.

HDR

Description

This command is identical to the HEADer query and is included for backward compatibility purposes.

Group

Miscellaneous

Related Commands

HEADer (see page 289), VERBose (see page 300)

Syntax 1

```
HDR {OFF|ON|<NR1>}
```

Syntax 2

```
HDR?
```

Arguments

- `OFF`

  This sets the Response Header Enable State to false. This causes the oscilloscope to omit headers on query responses, so that only the argument is returned.

- `ON`

  This sets the Response Header Enable State to true. This causes the oscilloscope to include headers on applicable query responses. You can then use the query response as a command.

- `<NR1>`

  A 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the oscilloscope to omit headers on query responses.

Example 1

```
HDR OFF
```

This command specifies that the oscilloscope omits headers on query responses, so that only the argument is returned.

Example 2

```
HDR?
```

This query might return `:HEADER 1`, indicating that the oscilloscope is including headers on applicable query responses.

HEADer

Description

This command sets or queries the Response Header Enable State that causes the oscilloscope 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); they never return headers.

Group

Miscellaneous

Related Commands

HDR (see page 288), VERBose (see page 300)

Syntax 1

```
HEADer {OFF|ON|<NR1>}
```

Syntax 2

```
HEADer?
```

Arguments

- `OFF`

  This sets the Response Header Enable State to false. This causes the oscilloscope to omit headers on query responses, so that only the argument is returned.

- `ON`

  This sets the Response Header Enable State to true. This causes the oscilloscope to include headers on applicable query responses. You can then use the query response as a command.

- `<NR1>`

  A 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the oscilloscope to omit headers on query responses.

Example 1

```
HEADer OFF
```

This command specifies that the oscilloscope omits headers on query responses, so that only the argument is returned.

Example 2

```
HEADer?
```

This query might return `:HEADER 1`, indicating that the oscilloscope is including headers on applicable query responses.

ID?

Description

This query-only command returns identifying information about the oscilloscope and related firmware.

Group

Miscellaneous

Related Commands

IDN? (see page 275)

Syntax

```
ID?
```

Example

```
ID?
```

This query might return `:TEK/TDS7104,CF:91.1CT,FV:01.00.912`, indicating the oscilloscope model number, configured format, and firmware version number.

LOCk

Description

This command enables or disables all front panel buttons and knobs, including the touch screen. There is no front panel equivalent.

Group

Miscellaneous

Related Commands

UNLock (see page 299)

Syntax 1

```
LOCk {ALL|NONe}
```

Syntax 2

```
LOCk?
```

Arguments

- `ALL`

  This disables all front panel controls.

- `NONe`

  This enables all front panel controls. This is equivalent to the UNLock ALL command.

  Note: If the oscilloscope 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.

Example 1

```
LOCk ALL
```

This command locks the front panel controls.

Example 2

```
LOCk?
```

This query might return `:LOCK NONE`, indicating that the front panel controls are enabled by this command.

NEWpass

Description

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

Related Commands

PASSWord (see page 293), *PUD (see page 324)

Syntax

```
NEWpass <QString>
```

Argument

- `<Qstring>`

  This is the new password, which can contain up to 10 characters.

Example

```
NEWpass "mypassword"
```

This command creates a new password for accessing your protected data.

PASSWord

Description

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 oscilloscope 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

Related Commands

NEWpass (see page 292), *PUD (see page 324)

Syntax

```
PASSWord <QString>
```

Argument

- `<QString>`

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

Example 1

```
PASSWord "XYZZY"
```

This command enables the *PUD and NEWPass set commands.

Example 2

```
PASSWord
```

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

REM

Description

This command (no query form) embeds a comment within GPIB programs as a means of internally documenting the programs. The oscilloscope ignores these embedded comment lines.

Group

Miscellaneous

Related Commands

NEWpass (see page 292), *PUD (see page 324)

Syntax

```
REM <QString>
```

Argument

- `<QString>`

  This is a string that can contain a maximum of 80 characters.

Example

```
REM "This is a comment"
```

The oscilloscope ignores this comment string.

SET?

Description

This query-only command returns the commands that list the oscilloscope settings, except for configuration information for the calibration values. You can use these commands to return the oscilloscope to the state it was in when you made the SET? query. This command is identical to the *LRN? command.

Group

Miscellaneous

Related Commands

HEADer (see page 289), *LRN? (see page 276), VERBose (see page 300)

Syntax

```
SET?
```

> **Note:** 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 oscilloscope as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full-length.

Example

```
SET?
```

This query might return the following response:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV
10;NUMAVG 16;REPET 1;:FASTACQ:STATE 1;:APPLICATION
:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;
:APPLICATION:WINDOW:HEIGHT 0;WIDTH 0;:APPLICATION
:SCOPEAPP:STATE NOTRUNNING;WINDOW FULLSCREEN;
:APPLICATION:EXTAPP:STATE NOTRUNNING;:AUXOUT:SOURCE
ATRIGGER;EDGE FALLING;:CMDBATCH 1;:HEADER 0;:LOCK
NONE;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1
;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR
DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE
FULL;INTENSITY:WAVEFORM 60.0000;AUTOBRIGHT 1
;SCREENSAVER 1;SCREENSAVERDELAY 28800;:DISPLAY
:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1
;VARPERSIST 500.0000E-3;:HARDCOPY:FILENAME "";FORMAT
BMP;LAYOUT PORTRAIT;PALETTE HARDCOPY;PORT FILE;
:HORIZONTAL:DELAY:MODE 1;POSITION 50.0000;TIME
0.0000;:HORIZONTAL:MAIN:SCALE 200.0000E-9;POSITION
50.0000;:HORIZONTAL:RECORDLENGTH 2500;ROLL AUTO
;FASTFRAME:STATE 0;LENGTH 2500;COUNT 2;SELECTED:CH1 2
;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2
;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME
:REF:SOURCE CH1;FRAME 1;:HORIZONTAL:FASTFRAME:TRACK
LIVE;:DIAG:LEVEL SUBSYS;:TRIGGER:A:MODE AUTO;TYPE
EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 250.0000E-9;
:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
: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: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: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 SLOWERTHAN;:TRIGGER:A:PULSE
:WIDTH:LOWLIMIT 2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN
;POLARITYPOSITIVE;:TRIGGER:A:PULSE:TIMEOUT:POLARITY
```

```
STAYSHIGH;TIME 2.0000E-9;: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;
:MATH1:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 65;:MATH1:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH2:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 80;:MATH2:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH3:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 95;:MATH3:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3;SPAN
600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6;WINDOW
GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0;
:MATH4:DEFINE " ";NUMAVG 2;SCALE 1.0000;POSITION 0.0000
;LABEL:NAME "";XPOS 5;YPOS 110;:MATH4:SPECTRAL:MAG DB
;PHASE DEGREES;GATEPOS -800.0000E-12;GATEWIDTH 1.9992E-6
;REFLEVEL 20.0000;REFLEVELOFFSET 223.6000E-3
;SPAN 600.0000E+6;CENTER 325.0000E+6;RESBW 1.0004E+6
;WINDOW GAUSSIAN;SUPPRESS -35000.0000E-3;UNWRAP 0;LOCK 0
;:HISTOGRAM:BOXPCNT 30.0000,25.1000,70.0000,75.2000
;DISPLAY LINEAR;STATE 0;FUNCTION HORIZONTAL;SIZE 2.0000
;SOURCE CH1;:CH1:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW
0.0000;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH1:LABEL:NAME "";XPOS 5;YPOS 5;
:CH2:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH2:LABEL:NAME "";XPOS 5;YPOS 20;
:CH3:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH3:LABEL:NAME "";XPOS 5;YPOS 35;
:CH4:BANDWIDTH 1.0000E+9;COUPLING DC;DESKEW 0.0000
;OFFSET 0.0000;POSITION 0.0000;SCALE 100.0000E-3
;TERMINATION 1.0000E+6;PROBEFUNC:EXTATTEN 1.0000
;EXTUNITS "None";:CH4:LABEL:NAME "";XPOS 5;YPOS 50;
: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;:CURSOR:STATE OFF;FUNCTION VBARS;MODE INDEPENDENT
;SOURCE CH1;VBARS:UNITS SECONDS;POSITION1 -800.0000E-9
;POSITION2 800.0000E-9;:CURSOR:HBARS:POSITION1
300.0000E-3;POSITION2 -300.0000E-3;:CURSOR:PAIRED
:POSITION1 -800.0000E-9;POSITION2 800.0000E-9;:CURSOR
:SPLIT:POSITION1 -800.0000E-9;POSITION2 800.0000E-9
;SOURCE2 CH1;UNITS BASE;MEASUREMENT:GATING OFF;METHOD
HISTOGRAM;IMMED:TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;SOURCE1 CH1
;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION
FORWARDS;:MEASUREMENT:MEAS2:STATE 0;TYPE UNDEFINED
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS3:STATE 0
;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1
RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS4
:STATE 0;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1
;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;
:MEASUREMENT:MEAS5:STATE 0;TYPE UNDEFINED;SOURCE1 CH1
;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION
FORWARDS;:MEASUREMENT:MEAS6:STATE 0;TYPE UNDEFINED
;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE
;DIRECTION FORWARDS;:MEASUREMENT:MEAS7:STATE 0;TYPE
UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE
```

```
;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS8
:STATE 0;TYPE UNDEFINED;SOURCE1 CH1;SOURCE2 CH1;DELAY
:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT
:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000
;MID1 0.0000;MID2 0.0000;:MEASUREMENT:REFLEVEL:PERCENT
:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;
:MEASUREMENT:STATISTICS:MODE OFF;WEIGHTING 32;:ZOOM
:MODE 0;GRATICULE:SIZE 50;:ZOOM:HORIZONTAL:LOCK ALL;
:ZOOM:CH1:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM
:CH1:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM:CH2
:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:CH2:VERTICAL
:POSITION 0.0000;SCALE 1.0000;:ZOOM:CH3:HORIZONTAL
:POSITION 50.0000;SCALE 2;:ZOOM:CH3:VERTICAL:POSITION
0.0000;SCALE 1.0000;:ZOOM:CH4:HORIZONTAL:POSITION
50.0000;SCALE 2;:ZOOM:CH4:VERTICAL:POSITION 0.0000
;SCALE 1.0000;:ZOOM:MATH1:HORIZONTAL:POSITION 50.0000
;SCALE 2;:ZOOM:MATH1:VERTICAL:POSITION 0.0000;SCALE
1.0000;:ZOOM:MATH2:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH2:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH3:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM
:MATH3:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM
:MATH4:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:MATH4
:VERTICAL:POSITION 0.0000;SCALE 1.0000;:ZOOM:REF1
:HORIZONTAL:POSITION 50.0000;SCALE 2;:ZOOM:REF1:VERTICAL
:POSITION 0.0000;SCALE 1.0000;:ZOOM:REF2:HORIZONTAL
:POSITION 50.0000;SCALE 2;:ZOOM:REF2:VERTICAL:POSITION
0.0000;SCALE 1.0000;:ZOOM:REF3:HORIZONTAL:POSITION
50.0000;SCALE 2;:ZOOM:REF3:VERTICAL:POSITION 0.0000
;SCALE 1.0000;:ZOOM:REF4:HORIZONTAL:POSITION 50.0000
;SCALE 2;:ZOOM:REF4:VERTICAL:POSITION 0.0000;SCALE
1.0000;:REF1:LABEL:NAME "";XPOS 5;YPOS 125;:REF1:SCALE
1.0000;POSITION 0.0000;:REF2:LABEL:NAME "";XPOS 5;YPOS
140;:REF2:SCALE 1.0000;POSITION 0.0000;:REF3:LABEL:NAME
"";XPOS 5;YPOS 155;:REF3:SCALE 1.0000;POSITION 0.0000;
:REF4:LABEL:NAME "";XPOS 5;YPOS 170;:REF4:SCALE 1.0000
;POSITION 0.0000;:DATA:DESTINATION REF1;ENCDG RIBINARY
;SOURCE CH1;START 1;STOP 500;WIDTH 8
```

TIMe

Description

This command sets or queries the time that the oscilloscope can display. 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

Related Commands

DATe (see page 287)

Syntax 1

```
TIMe <Qstring>
```

Syntax 2

```
TIMe?
```

Argument

- `<QString>`

  This 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.

Example 1

```
TIME "14:00:00"
```

This command specifies that the time is set to exactly 2:00 p.m..

Example 2

```
DATE?
```

This query might return `:TIME "14:05:17`, indicating the current time is set to 2:05 p.m..

UNLock

Description

This command (no query form) unlocks the front panel. The command is equivalent to LOCk NONe.

> **Note:** If the oscilloscope 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

Related Commands

LOCk (see page 291)

Syntax

```
UNLock ALL
```

Argument

- `ALL`

  This specifies that all front-panel buttons and knobs are unlocked.

Example

```
UNLock ALL
```

This command unlocks all front-panel buttons and knobs.

VERBose

Description

This command sets and queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).

Group

Miscellaneous

Related Commands

HEADer (see page 289), *LRN? (see page 276), SET? (see page 295)

Syntax

```
VERBose {OFF|ON|<NR1>}
```

Arguments

- `OFF`

  This sets the Verbose State to true, which returns full-length keywords for applicable setting queries.

- `ON`

  This 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.

Example 1

```
VERBose ON
```

This command enables the Verbose setting.

Example 2

```
VERBOSE?
```

This query might return `:VERBOSE 0`, indicating that the Verbose state is disabled.

Save and Recall Command Group

Save and Recall Overview

You 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 oscilloscope. When you recall a setup, the oscilloscope restores itself to the state that it was in when you originally saved that setting.

Save and Recall Commands

Command	Description
*RCL	Restores the state of the oscilloscope from a copy of the setting stored in memory.
*SAV	Stores the current oscilloscope state to a specified memory location
*SDS	Changes the specified setup to reference the factory setup instead of the user setup
DELEte:SETUp	Removes stored setups from memory and initializes the location with the factory default setups
DELEte:WAVEform	Deletes (one or all) of the stored reference waveforms from memory
FACtory	Resets the oscilloscope to the 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?	Returns the current format used for saving waveforms
SAVe:WAVEform:FILEFormat	Sets the format for saved waveforms
SETUp:NAMe?	Returns the user-defined setup label
SETUp:NAMe	Sets the user-defined setup label

*RCL

Description

This command (no query form) restores the state of the oscilloscope 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 Instrument Setup from the File menu and then choosing the Recall Setups tab.

Group

Save and Recall

Related Commands

DELEte:SETUp (see page 306), FACtory (see page 308), *LRN? (see page 276), RECAll:SETUp (see page 309), *RST (see page 325), *SAV (see page 304), SAVe:SETUp (see page 311)

Syntax

```
*RCL <NR1>
```

Argument

- `<NR1>`

  This specifies a setup storage location value, ranging from 0 through 10. Using an out-of-range value causes an execution error.

Example

```
*RCL 3
```

This command restores the oscilloscope from a copy of the settings stored in memory location 3.

*SAV

Description

This command (no query form) stores the state of the oscilloscope to a specified memory location. You can later use the *RCL command to restore the oscilloscope to this saved state. This is equivalent to selecting Instrument Setup from the File menu and then choosing the Save Setups tab.

Group

Save and Recall

Related Commands

*RCL (see page 303), RECAll:SETUp (see page 309), SAVe:SETUp (see page 311)

Syntax

```
*SAV <NR1>
```

Argument

- `<NR1>`

  This specifies a location in which to save the state of the scope. 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.

Example

```
*SAV 2
```

This command saves the current oscilloscope state in memory location 2.

*SDS

Description

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 Instrument Setups from the File menu, choosing the Recall Setups tab and then clicking the Default button.

Group

Save and Recall

Related Commands

DELEte:SETUp (see page 306)

Syntax

```
*SDS <NRf>
```

Argument

- `<NR1>`

  This specifies a user setup location that will be redirected. The setup in that position, which ranges from 1 through 10, will now refer to the Factory setup when recalled.

Example

```
*SDS 2
```

This command changes setup slot 2 to reference the factory setup.

DELEte:SETUp

Description

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 Instrument Setup from the File menu, choosing the Delete tab and then clicking the Delete button.

Note: The setup information cannot be recovered once it has been deleted.

Group

Save and Recall

Related Commands

*RCL? (see page 303), RECall:SETUp (see page 309), *RST (see page 325), *SAV (see page 304), SAVe:SETUp (see page 311), *SDS (see page 305)

Syntax

```
DELEte:SETUp {ALL|<NR1>}
```

Arguments

- `ALL`

  This specifies to delete all the stored setups.

- `<NR1>`

  This 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.

Example

```
DELEte:SETUp ALL
```

This command removes all stored setups. All ten storage locations are initialized to the factory default setup.

DELEte:WAVEform

Description

This command (no query form) deletes one or all stored reference waveforms from memory. This command is equivalent to selecting Reference Waveforms from the File menu and choosing Reference Setup from the drop-down list (when deleting individual reference waveforms) or selecting Delete All Refs from the File menu (when deleting all reference waveforms).

Group

Save and Recall

Related Commands

RECAll:WAVEform (see page 310), Save:WAVEform (see page 312)

Syntax

```
DELEte:WAVEform {ALL|REF<x>}
```

Arguments

- `ALL`

  This specifies to delete all the stored reference waveforms.

- `REF<x>`

  This specifies to delete one of the reference memory locations. Reference memory location values range from 1 through 4.

Example 1

```
DELEte:WAVEform ALL
```

This command removes all waveforms stored in reference memory.

Example 2

```
DELEte:WAVEform REF2
```

This command removes the waveform stored at REF2.

FACtory

Description

This command (no query form) resets the oscilloscope to factory default settings. This command is equivalent to pressing the **DEFAULT SETUP** button located on the front panel.

Group

Save and Recall

Related Commands

*PSC (see page 323), *RCL? (see page 303), RECall:SETUp (see page 309), *RST (see page 325), *SAV (see page 304), SAVe:SETUp (see page 311)

Syntax

```
FACtory
```

Arguments

None

Example

```
FACtory
```

This command resets the oscilloscope settings to factory defaults:

- Clears the Event Status Enable Register.
- Clears the Service request Enable Register.
- Sets the Device Event Status Enable Register to 2555.
- 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 implemented).

RECAll:SETUp

Description

This command (no query form) restores a stored or factory front-panel setup of the oscilloscope 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 Instrument Setup from the File menu and then choosing the Recall Setups tab.

Group

Save and Recall

Related Commands

FACtory (see page 308), *RCL (see page 303), *RST (see page 325), *SAV (see page 304), SAVe:SETUp (see page 311)

Syntax

```
RECAll:SETUp {FACtory|<NR1>|<file path>}
```

Arguments

- `FACtory`

  This specifies to restore the factory setup.

- `<NR1>`

  This 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>`

  This is the location from where the setup will be recalled.

  <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. <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.

Example 1

```
RECAll:SETUp FACtory
```

This command recalls (and makes current) the front-panel setup to its factory defaults.

Example 2

```
RECAll:SETUp 2
```

This command recalls the front panel setup from setup 2.

Example 3

```
RECAll:SETUp "TEK00000.SET"
```

This command recalls the front panel setup from the file `TEK00000.SET` in the default directory and on the default drive.

RECAll:WAVEform

Description

This command (no query form) recalls a stored waveform to a reference location. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Recall Wfm.

Group

Save and Recall

Related Commands

DELEte:WAVEform (see page 307), SAVe_WAVEform (see page 312)

Syntax

```
RECAll:WAVEform {<file path>|REF<x>}
```

Arguments

- `REF<x>`

  This specifies the location in internal reference memory from where the waveform is recalled. Reference memory location values range from 1 through 4.

- `<file path>`

  This is the location from where the waveform will be recalled.

  <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 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.

Example

```
RECAll:WAVEform "TEK00000.WFM",REF1
```

This command recalls the waveform stored in the file named TEK00000.WFM to reference location 1.

SAVe:SETUp

Description

This command (no query form) saves the current front-panel setup into the specified memory location or file. This is equivalent to selecting Instrument Setup from the File menu and then choosing the Save Setups tab.

Group

Save and Recall

Related Commands

*RCL (see page 303), RECAll:SETUp (see page 309),*SAV (see page 304)

Syntax

```
SAVe:SETUp {<file path>|<NR1>}
```

Arguments

- `<file path>`

  This is the location from where the waveform will be recalled.

  <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 "`SET`". The oscilloscope will generate an error if you use any other extension for saving a setup.

- `<NR1>`

  This 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.

Example 1

```
SAVe:SETUp 5
```

This command saves the current front-panel setup in memory location 5.

Example 2

```
SAVe:SETUp "TEK00000.SET"
```

This command saves the current front-panel setup in the file `TEK00000.SET` in the default directory and on the default drive.

SAVe:WAVEform

Description

This command (no query form) saves a waveform to one of four reference memory locations or a mass storage file. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Save Wfm from the drop-down list.

Group

Save and Recall

Related Commands

DELEte:WAVEform (see page 307), RECAll:WAVEform (see page 310)

Syntax

```
SAVe:WAVEform <wfm>,{<file path>|REF<x>}
```

Arguments

- `<wfm>`

  This is the waveform that will be saved. Valid waveforms include CH<x>, MATH<y>, and REF<x>.

- `REF<x>`

  This specifies the location in internal reference memory to where the waveform is stored.

- `<file path>`

  This 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 for spreadsheet format files or the .dat extension for MathCad format files. The oscilloscope will generate an error if you use any other extension for saving a waveform.

Example 1

```
SAVe:WAVEform MATH2,REF1
```

This command saves the Math2 waveform in reference memory location2.

Example 2

```
SAVe:WAVEform MATH1,"TEK0000.WFM"
```

This command saves the Math1 waveform to the file `TEK00000.WFM` in the default directory and on the default drive.

SAVe:WAVEform:FILEFormat

Description

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 500x200 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 value output.

This command is equivalent to selecting Export Setup from the File menu, clicking the Waveforms tab and then choosing the desired waveform file format from the Data destination drop-down list. Note that you choose the waveform file format after first selecting Select for Export from the File menu and then choosing Waveform (data) from the resulting drop-down list.

Group

Save and Recall

Related Commands

DATa:STARt (see page 473), DATa:STOP (see page 474)

Syntax 1

```
SAVe:WAVEform:FILEFormat {INTERNal|MATHCad|MATLab|SPREADSHEETCsv|SPREADSHEETTxt}
```

Syntax 2

```
SAVe:WAVEform:FILEFormat?
```

Arguments

- `INTERNal`

  This specifies that waveforms are saved in internal format, using a .wfm filename extension. These files can be recalled as reference waveforms.

- `MATHCad`

  This 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 file. MathCad format header files use a _hdr.dat extension.

- `MATLab`

  This 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 file. Matlab format header files use a _hdr.dat extension.

- `SPREADSHEETCsv`

  This 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`

  This 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.

Example 1

```
SAVe:WAVEform:FILEFormat INTERNal
```

This command specifies that the internal file format is the format used for saving waveforms.

Example 2

```
SAVe:WAVEform:FILEFormat?
```

This query might return: `SAVE:WAVEFORM:FILEFORMAT INTERNAL`, indicating that waveforms are saved using the internal format.

SETUp:NAMe

Description

This command sets and queries a Setup Label that you define. This command is equivalent to selecting Instrument Setup from the File menu, choosing the Save Setups tab and then adding a meaningful label to the selected setup.

Group

Save and Recall

Related Commands

SAVe:SETUp (see page 311)

Syntax 1

```
SETUp:NAMe <NR1>,<QString>
```

Syntax 2

```
SETUp:NAMe? <NR1>
```

Arguments

- `<NR1>`

  This specifies a location in which the Setup Label is stored. Location values range from 1 through 10.

- `<QString>`

  This is a string containing the Setup Label.

Example 1

```
SETUp:NAMe 1 "My Setup"
```

This command changes the Setup Label for internal setup slot 1 to "My Setup".

Example 2

```
SETUp:NAMe? 2
```

This query might return `:SETUP:NAME 2,"My Setup"`, indicating the Setup Label that you defined for internal setup slot 2.

Status and Error Command Group

Status and Error Overview

You use the commands in the Status and Error command Group to determine the status of the oscilloscope and control events.

Several commands and queries used with the oscilloscope 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.

Status and Error Commands

Command	Description
*CLS	Clears status
*ESE?	Returns the bits in the Event Status Enable Register
*ESE	Sets the bits in the Event Status Enable Register
*ESR?	Returns the contents of the Standard Event Status Register
*OPC?	Returns “1” when all current operations are finished
*OPC	Generates the operation complete message in the standard event status register when all pending operations are finished
*OPT?	Returns a list of options installed in the oscilloscope
*PSC?	Returns the power on status flag setting
*PSC	Sets the power on status flag
*PUD?	Returns a string of protected user data
*PUD	Sets a string of protected user data
*RST	Resets the oscilloscope to factory default settings
*SRE?	Returns the bits in the Service Request Enable Register
*SRE	Sets the bits in the Service Request Enable Register
*STB?	Returns the contents of the Status Byte Register
*WAI	Prevents the oscilloscope from executing further commands until all pending operations finish
ALLEv?	Returns all events and their messages
BUSY?	Returns oscilloscope status
DESE?	Returns the bits in the Device Event Status Enable Register
DESE	Sets the bits in the Device Event Status Enable Register
EVENT?	Returns event code from the event queue
EVMsg?	Returns event code, message from the event queue
EVQty?	Returns number of events in the event queue

*CLS

Description

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

Related Commands

DESE (see page 331), *ESE (see page 319), *ESR? (see page 320), EVENT? (see page 332), EVMsg? (see page 333), *SRE (see page 326), *STB? (see page 327)

Syntax

```
*CLS
```

Example

```
*CLS
```

This command clears the oscilloscope status data structures.

*ESE

Description

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 on page 522.

Group

Status and Error

Related Commands

*CLS (see page 318), DESE (see page 331), *ESR? (see page 320), EVENT? (see page 332), EVMsg? (see page 333), *SRE (see page 326), *STB? (see page 327)

Syntax 1

```
*ESE <NR1>
```

Syntax 2

```
*ESE?
```

Argument

- `<NR1>`

  This specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

  The power-on default for 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.

Example 1

```
*ESE 209
```

This command sets the ESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.

Example 2

```
*ESE?
```

This query form of this command returns `186`, showing that the ESER contains the binary value 10111010.

*ESR?

Description

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 on page 522.

Group

Status and Error

Related Commands

ALLEv? (see page 329), *CLS (see page 318), DESE (see page 331), *ESE (see page 319), EVENT? (see page 332), EVMsg? (see page 333), *SRE (see page 326), *STB? (see page 327)

Syntax

```
*ESR?
```

Example

```
*ESR?
```

This query might return 213, showing that the SESR contains the binary value 11010101.

*OPC

Description

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending operations finish. The *OPC? query places the ASCII character "1" into the output queue when all pending operations are finished. 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 on page 522 and Queues on page 527.

The *OPC command allows you to synchronize the operation of the oscilloscope with your application program. For more information, see Synchronization Methods on page 542.

Commands that Generate an OPC Message

Operation	Command
Single sequence acquisition	ACQuire:STATE ON *or* ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQuence)

Group

Status and Error

Related Commands

BUSY? (see page 330), *WAI (see page 328)

Syntax 1

```
*OPC
```

Syntax 2

```
*OPC?
```

Example 1

```
*OPC
```

Upon completion of all pending operations, this command generates the operation complete message in the SESR.

Example 2

```
*OPC?
```

This query might return `1` to indicate that all pending operations are finished.

*OPT?

Description

This query-only command returns a list of the options installed in your oscilloscope.

Group

Status and Error

Syntax

```
*OPT?
```

Example

```
*OPT?
```

This query might return `:3M:16 Meg Max 4M/CH`

*PSC

Description

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

Related Commands

DESE (see page 331), *ESE (see page 319), FACtory (see page 308), *RST (see page 325), *SRE (see page 326)

Syntax 1

```
*PSC <NR1>
```

Syntax 2

```
*PSC?
```

Argument

- `<NR1>`

  A 0 sets the power-on status clear flag to false, disables the power-on clear and allows the oscilloscope 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.

Example 1

```
*PSC 0
```

This command sets the power-on status clear flag to false.

Example 2

```
*PSC?
```

The query form of this command might return `1` to indicate that the power-on status clear flag is set to true.

*PUD

Description

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

Related Commands

PASSWord (see page 293)

Syntax 1

```
*PUD {<Block>|<String>}
```

Syntax 2

```
*PUD?
```

Argument

- `<Block>`

  This is a block containing up to 100 characters.

- `<String>`

  This is a string containing up to 100 characters.

Example 1

```
*PUD #229This oscilloscope belongs to me
```

This command stores the string "This oscilloscope belongs to me" in the user protected data area.

Example 2

```
*PUD?
```

This query might return `#221Property of Company X`

*RST

Description

This command (no query form) resets the oscilloscope 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 oscilloscope
- 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

This command is equivalent to pressing the **DEFAULT SETUP** button on the front-panel.

Group

Status and Error

Related Commands

FACtory (see page), RECAll:SETUp (see page 309), SAVe:SETUp (see page 311)

Syntax

```
*RST
```

Argument

None.

Example

```
*RST
```

This command resets the oscilloscope settings to factory defaults.

*SRE

Description

The *SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers on page 522.

Group

Status and Error

Related Commands

*CLS (see page 318), DESE (see page 331), *ESE (see page 319), *ESR? (see page 320), EVENT? (see page 332), EVMSg? (see page 333), FACtory (see page 308), *STB? (see page 327)

Syntax 1

```
*SRE <NR1>
```

Syntax 2

```
*SRE?
```

Argument

- `<NR1>`

  This 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 throughout the current power cycle.

Example 1

```
*SRE 48
```

This command sets the bits in the SRER to 00110000 binary.

Example 2

```
*SRE?
```

This query might return `32`, showing that the bits in the SRER have the binary value of 00100000.

*STB?

Description

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 on page 522.

Group

Status and Error

Related Commands

CLS (see page 318), DESE (see page 331), *ESE (see page 319), *ESR? (see page 320), EVENT? (see page 332), EVMSg? (see page 333), FACtory (see page 308), *SRE? (see page 326)

Syntax

```
*STB?
```

Returns

```
<NR1>
```

Example

```
*STB?
```

This query might return `96`, showing that the SBR contains the binary value 01100000.

*WAI

Description

The *WAI (Wait) command (no query form) prevents the oscilloscope from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the oscilloscope with your application program. For more information, refer to Synchronization Methods on page 542.

Group

Status and Error

Related Commands

BUSY? (see page 330), *OPC (see page 321)

Syntax

```
*WAI
```

Example

```
*WAI
```

This command prevents the oscilloscope from executing any further commands or queries until all pending operations finish.

ALLEV?

Description

This query-only command prompts the oscilloscope 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 oscilloscope.

Group

Status and Error

Related Commands

*ESR? (see page 320), *EVMsg? (see page 333)

Syntax

```
ALLev?
```

Example

```
ALLev?
```

This query might return `:ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UNTERMINATED;"`

BUSY?

Description

This query-only command returns the status of the oscilloscope. This command allows you to synchronize the operation of the oscilloscope with your application program.

Group

Status and Error

Related Commands

*OPC (see page 321), *WAI (see page 328)

Syntax

```
BUSY?
```

Returns

```
<NR1>
```

If 0 is returned, it means that the oscilloscope is not busy processing a command whose execution time is extensive. If 1 is returned, it means that the oscilloscope 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 (when ACQuire:STOPAfter is set to SEQuence)

Example

```
BUSY?
```

This query might return `:BUSY 1`, indicating that the oscilloscope is currently busy.

DESE

Description

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 on page 522.

Group

Status and Error

Related Commands

*CLS (see page 318), *ESE (see page 319), *ESR? (see page 320), *EVENT? (see page 332), EVMsg? (see page 333), *SRE (see page 326), *STB? (see page 327)

Syntax 1

```
DESE <NR1>
```

Syntax 2

```
DESE?
```

Argument

- `<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.

Example 1

```
DESE 209
```

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

Example 2

```
DESE?
```

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

EVENT?

Description

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

Related Commands

ALLEv? (see page 329), *CLS (see page 318), DESE (see page 331), *ESE (see page 319), *ESR? (see page), EVMsg? (see page 333), *SRE (see page 326), *STB? (see page 327)

Syntax

```
EVENT?
```

Example

```
EVENT?
```

This query might return `:EVENT 110`, showing that there was an error in a command header.

EVMsg?

Description

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 on page 530.

Group

Status and Error

Related Commands

ALLEv? (see page 329), *CLS (see page 318), DESE (see page 331), *ESE (see page 319), *ESR? (see page), EVENT? (see page 332), *SRE (see page 326), *STB? (see page 327)

Syntax

```
EVMsg?
```

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 oscilloscope. 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.

Example

```
EVMsg?
```

This query might return `:EVMSG 110, "Command header error"`

EVQty?

Description

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

Related Commands

ALLEv? (see page 329), EVENT? (see page 332), EVMsg? (see page 333)

Syntax

```
EVQty?
```

Example

```
EVQty?
```

This query might return `:EVQTY 3`, indicating the number of event codes in the Event Queue.

Trigger Command Group

Trigger Overview

You use the commands in the Trigger Command Group to control all aspects of triggering for the oscilloscope.

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 triggers to pulse and logic modes. Pulse triggering lets the oscilloscope trigger 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 oscilloscope then triggers when it detects a certain combination of signal levels.

Trigger Commands

Command	Description
TRIGger?	Returns current trigger parameters for the oscilloscope
TRIGger	Forces a trigger event to occur
TRIGger:A?	Returns current A trigger parameters
TRIGger:A	Sets A trigger level to 50%
TRIGger:A:EDGE?	Returns the source, coupling and source for the A edge trigger
TRIGger:A:EDGE:COUPling?	Returns the type of coupling for the A edge trigger
TRIGger:A:EDGE:COUPling	Sets the type of coupling for the A edge trigger
TRIGger:A:EDGE:SLOpe?	Returns the slope for the A edge trigger
TRIGger:A:EDGE:SLOpe	Sets the slope for the A edge trigger
TRIGger:A:EDGE:SOUrce?	Returns the source for the A edge trigger
TRIGger:A:EDGE:SOUrce	Sets the source for the A edge trigger
TRIGger:A:HOLDoff?	Returns the A trigger holdoff parameters
TRIGger:A:HOLDoff:ACTUal?	Returns the holdoff time actually used by the A trigger
TRIGger:A:HOLDoff:BY?	Returns the type of holdoff for the A trigger
TRIGger:A:HOLDoff:BY	Sets the type of holdoff for the A trigger
TRIGger:A:HOLDoff:TIMe?	Returns the A trigger holdoff time
TRIGger:A:HOLDoff:TIMe	Sets the A trigger holdoff time
TRIGger:A:LEVel?	Returns the level for the A trigger
TRIGger:A:LEVel	Sets the level for the A trigger
TRIGger:A:LOGIc?	Returns all of the A trigger logic parameters
TRIGger:A:LOGIc:CLAss?	Returns the type of A trigger logic
TRIGger:A:LOGIc:CLAss	Sets the type of A trigger logic
TRIGger:A:LOGIc:FUNCtion?	Returns the logical combination of the input

	channels for the A logic trigger
TRIGger:A:LOGIc:FUNCtion	Sets the logical combination of the input channels for the A logic trigger
TRIGger:A:LOGIc:INPut?	Returns the A logic trigger input for channels 1-3.
TRIGger:A:LOGIc:INPut:CH<x>?	Returns the input settings for the specified logic trigger channel
TRIGger:A:LOGIc:INPut:CH<x>	Specifies the input settings for the specified logic trigger channel
TRIGger:A:LOGIc:PATtern?	Returns the conditions for generating an A logic pattern trigger
TRIGger:A:LOGIc:PATtern:INPut:CH4?	Returns the A logic trigger input for channel 4
TRIGger:A:LOGIc:PATtern:INPut:CH4	Sets the A logic trigger input for channel 4
TRIGger:A:LOGIc:PATtern:WHEn?	Returns the condition for generating the A logic pattern trigger
TRIGger:A:LOGIc:PATtern:WHEn	Sets the condition for generating the A logic pattern trigger
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit?	Returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGIc:PATtern:WHEn:LEssLimit	Sets the maximum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit?	Returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit	Sets the minimum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGIc:SETHold?	Returns clock edge polarity, voltage threshold and source; data voltage threshold and source; and setup/hold times for violation triggering
TRIGger:A:LOGIc:SETHold:CLOCk?	Returns clock edge polarity, voltage threshold and source input for setup/hold triggering
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE?	Returns the clock edge polarity for setup and hold triggering
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE	Sets the clock edge polarity

	for setup and hold triggering
TRIGger:A:LOGIc:SETHold:CLOCk :SOUrce?	Returns the clock source for the A logic trigger setup and hold input
TRIGger:A:LOGIc:SETHold:CLOCk :SOUrce	Sets the clock source for the A logic trigger setup and hold input
TRIGger:A:LOGIc:SETHold:CLOCk :THReshold?	Returns the clock voltage threshold for setup and hold trigger
TRIGger:A:LOGIc:SETHold:CLOCk :THReshold	Sets the clock voltage threshold for setup and hold trigger
TRIGger:A:LOGIc:SETHold:DATa?	Returns the voltage threshold and data source for the setup/hold trigger
TRIGger:A:LOGIc:SETHold:DATa:SOUrce?	Returns the data source for the setup and hold trigger
TRIGger:A:LOGIc:SETHold:DATa:SOUrce	Sets the data source for the setup and hold trigger
TRIGger:A:LOGIc:SETHold:DATa :THReshold?	Returns the data voltage threshold for setup and hold trigger
TRIGger:A:LOGIc:SETHold:DATa :THReshold	Sets the data voltage threshold for setup and hold trigger
TRIGger:A:LOGIc:SETHold:HOLDTime?	Returns the hold time for the setup and hold violation triggering
TRIGger:A:LOGIc:SETHold:HOLDTime	Sets the hold time for the setup and hold violation triggering
TRIGger:A:LOGIc:SETHold:SETTime?	Returns the setup time for setup and hold violation triggering
TRIGger:A:LOGIc:SETHold:SETTime	Sets the setup time for setup and hold violation triggering
TRIGger:A:LOGIc:STATE?	Returns the data input and trigger criteria for the A logic trigger
TRIGger:A:LOGIc:STATE:INPut:CH4?	Returns the slope for channel 4 of the A logic state trigger
TRIGger:A:LOGIc:STATE:INPut:CH4	Sets the slope for channel 4 of the A logic state trigger
TRIGger:A:LOGIc:STATE:WHEn?	Returns the condition for generating an A logic state trigger
TRIGger:A:LOGIc:STATE:WHEn	Sets the condition for generating an A logic state trigger
TRIGger:A:LOGIc:THReshold?	Returns the threshold voltage for all channels in A logic trigger

TRIGger:A:LOGIc:THReshold:CH<x>?	Returns the A logic trigger threshold voltage for the specified channel
TRIGger:A:LOGIc:THReshold:CH<x>	Sets the A logic trigger threshold voltage for the specified channel
TRIGger:A:MODe?	Returns the A trigger mode
TRIGger:A:MODe	Sets the A trigger mode
TRIGger:A:PULse?	Returns the A pulse trigger parameters
TRIGger:A:PULse:CLAss?	Returns the type of pulse on which to trigger
TRIGger:A:PULse:CLAss	Sets the type of pulse on which to trigger
TRIGger:A:PULse:GLItch?	Returns the current A glitch pulse trigger parameters
TRIGger:A:PULse:GLItch:POLarity?	Returns the polarity for the A pulse glitch trigger
TRIGger:A:PULse:GLItch:POLarity	Sets the polarity for the A pulse glitch trigger
TRIGger:A:PULse:GLItch:TRIGIF?	Returns the acceptance or rejection of pulse glitch trigger, based on width.
TRIGger:A:PULse:GLItch:TRIGIF	Sets the acceptance or rejection of pulse glitch trigger, based on width.
TRIGger:A:PULse:GLItch:WIDth?	Returns the width for the A pulse glitch trigger
TRIGger:A:PULse:GLItch:WIDth	Sets the width of the A pulse glitch trigger
TRIGger:A:PULse:RUNT?	Returns the current A runt pulse trigger parameters
TRIGger:A:PULse:RUNT:POLarity?	Returns the polarity for the A pulse runt trigger
TRIGger:A:PULse:RUNT:POLarity	Sets the polarity for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold?	Returns the upper and lower thresholds for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:BOTh	Sets the upper and lower switching thresholds for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:HIGH?	Returns the upper limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:HIGH	Sets the upper limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:LOW?	Returns the lower limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:LOW	Sets the lower limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:WHEn?	Returns the type of pulse

	width the trigger checks for when it uncovers a runt
TRIGger:A:PULse:RUNT:WHEn	Sets the type of pulse width the trigger checks for when it uncovers a runt
TRIGger:A:PULse:RUNT:WIDth?	Returns the minimum width for A pulse runt trigger
TRIGger:A:PULse:RUNT:WIDth	Sets the minimum width for A pulse runt trigger
TRIGger:A:PULse:SOURce?	Returns the source for the A pulse trigger
TRIGger:A:PULse:SOURce	Sets the source for the A pulse trigger
TRIGger:A:PULse:TIMEOut?	Returns the parameters for the A pulse timeout trigger
TRIGger:A:PULse:TIMEOut:POLarity?	Returns the polarity for the A pulse timeout trigger
TRIGger:A:PULse:TIMEOut:POLarity	Sets the polarity for the A pulse timeout trigger
TRIGger:A:PULse:TIMEOut:TIMe?	Returns the pulse timeout trigger time
TRIGger:A:PULse:TIMEOut:TIMe	Sets the pulse timeout trigger time
TRIGger:A:PULse:TRANsition?	Returns the delta time, polarity, and both upper and lower threshold limits for the transition time trigger
TRIGger:A:PULse:TRANsition:DeltaTime?	Returns the delta time used in calculating the transition value
TRIGger:A:PULse:TRANsition:DeltaTime	Sets the delta time used in calculating the transition value
TRIGger:A:PULse:TRANsition:POLarity?	Returns the polarity for the A pulse transition trigger
TRIGger:A:PULse:TRANsition:POLarity	Sets the polarity for the A pulse transition trigger
TRIGger:A:PULse:TRANsition:THReshold?	Returns the upper and lower threshold limits for the transition time trigger
TRIGger:A:PULse;TRANsition:THReshold:BOTh	Sets the upper and lower thresholds for the pulse transition trigger
TRIGger:A:PULse:TRANsition:THReshold:HIGH?	Returns the upper transition trigger threshold
TRIGger:A:PULse:TRANsition:THReshold:HIGH	Sets the upper transition trigger threshold
TRIGger:A:PULse:TRANsition:THReshold:LOW?	Returns the lower transition trigger threshold
TRIGger:A:PULse:TRANsition:THReshold:LOW	Sets the lower transition trigger threshold

TRIGger:A:PULse:TRANsition:WHEn?	Returns the relationship of delta time to transitioning signal
TRIGger:A:PULse:TRANsition:WHEn	Sets the relationship of delta time to transitioning signal
TRIGger:A:PULse:WIDth?	Returns the trigger A pulse width parameters
TRIGger:A:PULse:WIDth:HIGHLimit?	Returns the upper limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:HIGHLimit	Sets the upper limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:LOWLimit?	Returns the lower limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:LOWLimit	Sets the lower limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:POLarity?	Returns the polarity for the A pulse width trigger
TRIGger:A:PULse:WIDth:POLarity	Sets the polarity for the A pulse width trigger
TRIGger:A:PULse:WIDth:WHEn?	Returns the criteria for width specification of pulse width trigger events
TRIGger:A:PULse:WIDth:WHEn	Sets the criteria for width specification of pulse width trigger events
TRIGger:STATE?	Returns the current state of the triggering system
TRIGger:A:TYPe?	Returns the type of A trigger
TRIGger:A:TYPe	Sets the type of A trigger
TRIGger:B?	Returns the B trigger parameters
TRIGger:B	Sets the B trigger level to 50%
TRIGger:B:BY?	Returns B trigger delay qualifiers
TRIGger:B:BY	Sets B trigger time or event qualifiers
TRIGger:B:EDGE?	Returns B trigger edge type parameters
TRIGger:B:EDGE:COUPling?	Returns the type of B trigger coupling
TRIGger:B:EDGE:COUPling	Sets the type of B trigger coupling
TRIGger:B:EDGE:SLOpe?	Returns the B edge trigger slope
TRIGger:B:EDGE:SLOpe	Sets the B edge trigger slope
TRIGger:B:EDGE:SOUrce?	Returns the B edge trigger source
TRIGger:B:EDGE:SOUrce	Sets the B edge trigger source

TRIGger:B:EVENTS?	Returns the current B trigger events parameter
TRIGger:B:EVENTS:COUNt?	Returns the number of events that must occur before the B trigger occurs
TRIGger:B:EVENTS:COUNt	Sets the number of events that must occur before the B trigger occurs
TRIGger:B:LEVel?	Returns the level for the B trigger
TRIGger:B:LEVel	Sets the level for the B trigger
TRIGger:B:STATE?	Returns the state of the B trigger
TRIGger:B:STATE	Sets the state of the B trigger
TRIGger:B:TIMe?	Returns the B trigger delay time
TRIGger:B:TIMe	Sets the B trigger delay time
TRIGger:B:TYPe?	Returns the type of B trigger
TRIGger:B:TYPe	Sets the type of B trigger

TRIGger

Description

This command forces a trigger event to occur and the query returns current trigger parameters for the oscilloscope.

Group

Trigger

Syntax 1

```
TRIGger FORCe
```

Syntax 2

```
TRIGger?
```

Argument

- `FORCe`

  This 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.

Example 1

```
TRIGger FORCe
```

This command forces a trigger event to occur.

Example 2

```
TRIGger?
```

This query might return

```
:TRIGGER:A:MODE AUTO;TYPE EDGE
;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 250.0000E-9;
:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
: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: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: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 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: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

Description

This command sets the A trigger level automatically to 50% of the minimum/maximum value of the trigger input signal. The query returns current A trigger parameters. TRIGger:A? is a composite of TRIGger:A:EDGE?, TRIGger:A:LOGIc?, and TRIGger:A:PULSE? 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

Related Commands

TRIGger:A:EDGE? (see page 345), TRIGger:A:LOGIc? (see page 354), TRIGger:A:PULSE? (see page 380)

Syntax 1

```
TRIGger:A SETLevel
```

Syntax 2

```
TRIGger:A?
```

Argument

- `SETLevel`

  This sets the A trigger level to 50% of minimum and maximum.

Example 1

```
TRIGger:A SETLevel
```

This command sets the A trigger level to 50% of minimum and maximum.

Example 2

```
TRIGger:A?
```

This query might return
```
:TRIGGER:A:MODE AUTO;TYPE EDGE
;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 250.0000E-9;
:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
: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: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: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 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:EDGE?

Description

This query-only command returns the trigger source, coupling and slope for the A edge trigger. This command is equivalent to selecting Edge Setup from the Trig menu and then viewing the current setups.

Group

Trigger

Related Commands

TRIGger:A:PULse? (see page 380), TRIGger:A:LOGIc? (see page 354)

Syntax

```
TRIGger:A:EDGE?
```

Example

```
TRIGger:A:EDGE?
```

This query 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:EDGE:COUPling

Description

This command sets or queries the type of coupling for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing from the Coupling drop-down list.

Group

Trigger

Related Commands

TRIGger:A:EDGE:SOUrce (see page 348), TRIGger:A:EDGE:SLOpe (see page 347)

Syntax 1

```
TRIGger:A:EDGE:COUPling {AC|DC|HFRej|LFRej|NOISErej}
```

Syntax 2

```
TRIGger:A:EDGE:COUPling?
```

Arguments

- `AC`

  This selects AC trigger coupling.

- `DC`

  This selects DC trigger coupling.

- `HFRej`

  This coupling removes the high frequency components of the DC signal.

- `LFRej`

  This coupling removes the low frequency components of the AC signal.

- `NOISErej`

  This selects DC low sensitivity. It requires added signal amplitude for more stability and less false triggering.

Example 1

```
TRIGger:A:EDGE:COUPling DC
```

This command sets the A edge trigger coupling to DC.

Example 2

```
TRIGger:A:EDGE:COUPling?
```

This query might return `:TRIGGER:A:EDGE:COUPLING DC`, indicating that the A edge trigger coupling is set to DC.

TRIGger:A:EDGE:SLOpe

Description

This command sets or queries the slope for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired Slope.

Group

Trigger

Related Commands

TRIGger:A:EDGE:SOUrce (see page 348), TRIGger:A:EDGE:COUPling (see page 346)

Syntax 1

```
TRIGger:A:EDGE:SLOpe {RISe|FALL}
```

Syntax 2

```
TRIGger:A:EDGE:SLOpe?
```

Arguments

- `RISe`

  This specifies to trigger on the rising or positive edge of a signal.

- `FALL`

  This specifies to trigger on the falling or negative edge of a signal.

Example 1

```
TRIGger:A:EDGE:SLOpe?
```

This query might return `:TRIGGER:A:EDGE:SLOPE FALL`, indicating that the A edge trigger slope is negative.

Example 2

```
TRIGger:A:EDGE:SLOpe RISe
```

This command sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

TRIGger:A:EDGE:SOUrce

Description

This command sets or queries the source for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing from the Source drop-down list.

Group

Trigger

Related Commands

TRIGger:A:EDGE:SLOpe (see page 347), TRIGger:A:EDGE:COUPling (see page 346)

Syntax 1

```
TRIGger:A:EDGE:SOUrce {AUXiliary|CH<x>|LINE}
```

Syntax 2

```
TRIGger:A:EDGE:SOUrce?
```

Arguments

- `AUXiliary`

  This specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the oscilloscope.

- `CH<x>`

  This specifies one input channel as the A edge trigger source.

- `LINE`

  This specifies AC line voltage.

Example 1

```
TRIGger:A:EDGE:SOUrce?
```

This query might return `:TRIGGER:A:EDGE:SOURCE CH1`, indicating that channel 1 is the A edge trigger source.

Example 2

```
TRIGger:A:EDGE:SOUrce CH1
```

This command specifies channel 1 as the A edge trigger source.

TRIGger:A:HOLDoff?

Description

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

Related Commands

TRIGger:A:HOLDoff:ACTUal? (see page 350), TRIGger:A:HOLDoff:BY (see page 351), TRIGger:A:HOLDoff:TIMe (see page 352)

Syntax

```
TRIGger:A:HOLDoff?
```

Example

```
TRIGger:A:HOLDoff?
```

This query might return `:TRIGGER:A:HOLDOFF:TIME 900.0000E-09;BY DEFAULT`, indicating that the A edge trigger holdoff time is set 900 ns.

TRIGger:A:HOLDoff:ACTUal?

Description

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

Related Commands

TRIGger:A:HOLDoff? (see page 349), TRIGger:A:HOLDoff:BY (see page 351), TRIGger:A:HOLDoff:TIMe (see page 352)

Syntax

```
TRIGger:A:HOLDoff:ACTUal?
```

Example

```
TRIGger:A:HOLDoff:ACTUal?
```

This query might return `:TRIGGER:A:HOLDOFF:ACTUAL 4.0000E-06`, showing that the holdoff time is set to 4 µs.

TRIGger:A:HOLDoff:BY

Description

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). This command is equivalent to selecting Holdoff from the Trig menu and then viewing or setting the Holdoff type.

Group

Trigger

Related Commands

TRIGger:A:HOLDoff:TIMe (see page 352)

Syntax 1

```
TRIGger:A:HOLDoff:BY {TIMe|DEFAult}
```

Syntax 2

```
TRIGger:A:HOLDoff:BY?
```

Arguments

- `TIMe`

  This enables you to set the holdoff time via the TRIGger:A:HOLDoff:TIMe command.

- `DEFAult`

  This 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 oscilloscope is set to 1 ms/division then the default holdoff will be 1 ms/division x 25 divisions = 25 ms.

Example 1

```
TRIGger:A:HOLDoff:BY?
```

This query might return `:TRIGGER:A:HOLDOFF:BY TIME`, indicating that you will set the holdoff time.

Example 2

```
TRIGger:A:HOLDoff:BY TIMe
```

This command sets the holdoff to the "by time" setting. This enables you to set the holdoff time.

TRIGger:A:HOLDoff:TIMe

Description

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

Related Commands

TRIGger:A:HOLDoff:BY (see page 351)

Syntax 1

```
TRIGger:A:HOLDoff:TIMe <NR3>
```

Syntax 2

```
TRIGger:A:HOLDoff:TIMe?
```

Argument

- `<NR3>`

  This specifies the holdoff time in seconds. The range is from 250 ns through 12.0 s.

Example 1

```
TRIGger:A:HOLDoff:TIME?
```

This query might return `:TRIGGER:A:HOLDOFFTIME 1.2000E-06`, indicating that the A trigger holdoff time is set to 1.2 µs.

Example 2

```
TRIGger:A:HOLDoff:TIMe 10
```

This command sets the trigger A holdoff time to 10 s.

TRIGger:A:LEVel

Description

This command sets or queries the level for the A trigger. This command is equivalent to selecting Holdoff from the Trig menu and then viewing or setting the trigger Level.

Group

Trigger

Syntax 1

```
TRIGger:A:LEVel {ECL|TTL|<NR3>}
```

Syntax 2

```
TRIGger:A:LEVel?
```

Arguments

- `ECL`

  This specifies the ECL high level.

- `TTL`

  This specifies the TTL high level.

- `<NR3>`

  This specifies the trigger level in user units (usually volts).

Example 1

```
TRIGger:A:LEVel?
```

This query might return `:TRIGGER:A:LEVel 1.3000E+00`, indicating that the A edge trigger is set to 1.3 V.

Example 2

```
TRIGger:A:LEVel TTL
```

This command sets the A edge trigger to TTL high level, which is 1.4 V.

TRIGger:A:LOGIc?

Description

This query-only command returns all of the A logic trigger parameters.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:CLAss (see page 355)

Syntax

```
TRIGger:A:LOGic?
```

Example

```
TRIGger:A:LOGic?
```

This query 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:HOLDTIME 2.0000E-9;SETTIME 3.0000E-9;:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE;:TRIGGER :A:LOGIC:STATE:WHEN TRUE`

TRIGger:A:LOGIc:CLAss

Description

This command sets or queries the class of the Logic Trigger. Used in concert with TRIGger:A:TYPe, this command is equivalent to selecting Logic Pattern, Logic State or Setup/Hold Setup from the Trig menu.

Group

Trigger

Related Commands

TRIGger:A:TYPe (see page 412), TRIGger:A:PULse:CLAss (see page 381)

Syntax 1

```
TRIGger:A:LOGIc:CLAss {PATtern|STATE|SETHold}
```

Syntax 2

```
TRIGger:A:LOGIc:CLAss?
```

Arguments

- `PATtern`

  This means that the oscilloscope triggers when the specified logical combinations of channels 1, 2, 3 and 4 are met.

- `STATE`

  This means that the oscilloscope triggers when the specified conditions of channels 1, 2, and 3 are met after the channel 4 condition is met.

- `SETHold`

  This means that the oscilloscope will trigger on the setup and hold violations between a data source and a clock source.

Example 1

```
TRIGger:A:TYPe LOGIc
TRIGger:A:LOGIc:CLAss?
```

This query might return `:TRIGGER:A:LOGIC:CLASS PATTERN`

Example 2

```
TRIGger:A:LOGIc:CLAss PATTERN
```

This command sets the trigger A logic class to PATtern, which causes the oscilloscope to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met.

TRIGger:A:LOGIc:FUNCtion

Description

This command sets or queries the logical combination of the input channels for the A pattern and A state logic triggers. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then setting or viewing the Pattern Type.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:PATtern:WHEn (see page 361), TRIGger:A:LOGIc:INPut:CH<x> (see page 358), TRIGger:A:LOGIc:THReshold:CH<x> (see page 378)

Syntax 1

```
TRIGger:A:LOGIc:FUNCtion {AND|NANd|NOR|OR}
```

Syntax 2

```
TRIGger:A:LOGIc:FUNCtion?
```

Arguments

- `AND`

  This specifies to trigger if all conditions are true.

- `NANd`

  This specifies to trigger if any of the conditions are false.

- `NOR`

  This specifies to trigger if all conditions are false.

- `OR`

  This specifies to trigger if any of the conditions are true.

Example 1

```
TRIGger:A:LOGIc:FUNCTion?
```

This query might return `:TRIGGER:A:LOGIC:FUNCTION NAND`

Example 2

```
TRIGger:A:LOGIc:FUNCTion AND
```

This command sets the logical combination of channels to be true when all conditions are true.

TRIGger:A:LOGIc:INPut?

Description

This query-only command returns the A logic trigger input expected for channel 1, 2, and 3. Channel 4 is set or queried with the command TRIGger:A:LOGIc:PATtern:INPut:CH4. This command is equivalent to selecting A Event (Main) Trigger Setup and then viewing or setting the Input Threshold for channel 1 through 3.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:PATtern:INPut:CH4 (see page 360)

Syntax

```
TRIGger:A:LOGIc:INPut?
```

Example

```
TRIGger:A:LOGIc:INPut?
```

This query might return `:TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X`

TRIGger:A:LOGIc:INPut:CH<x>

Description

This command sets or queries the A logical input for the specified logic trigger channel, which ranges from CH1 through CH3. Note that CH4 cannot be set or queried with this command. For details about setting this channel, see TRIGger:A:LOGIc:PATtern:INPut:CH4. This command is equivalent to selecting A Event (Main) 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

Related Commands

TRIGger:A:LOGIc:PATtern:INPut:CH4 (see page 360)

Syntax 1

```
TRIGger:A:LOGIc:INPut:CH<x> {HIGH|LOW|X}
```

Syntax 2

```
TRIGger:A:LOGIc:INPut:CH<x>?
```

Arguments

- `HIGH`

  This specifies the logic high.

- `LOW`

  This specifies the logic low.

- `X`

  This specifies a "do not care" state.

Example 1

```
TRIGger:A:LOGIc:INPut:CH1?
```

This query might return `:TRIGGER:LOGIC:INPUT:CH1 X`, indicating that the setting for the A logic trigger input to channel 1 does not matter.

Example 2

```
TRIGger:A:LOGIc:INPut:CH2 HIGH
```

This command sets the A logic trigger input to logic HIGH for channel 2.

TRIGger:A:LOGIc:PATtern?

Description

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

Related Commands

TRIGger:A:LOGIc:PATtern:INPut:CH4 (see page 360), TRIGger:A:LOGIc:PATtern:WHEn (see page 361), TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit (see page 362), TRIGger:A:LOGIc:PATtern:WHEn:MORELimit (see page 363)

Syntax

```
TRIGger:A:LOGIc:PATtern?
```

Example

```
TRIGger:A:LOGIc:PATtern?
```

This query 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:LOGIc:PATtern:INPut:CH4

Description

This command sets or queries the A logic trigger input for channel 4. 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 Ch4 drop-down list, which is located in the Input Threshold group box.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:FUNCtion (see page 356), TRIGger:A:LOGIc:INPut:CH<x> (see page 358), TRIGger:A:LOGIc:THReshold:CH<x> (see page 378)

Syntax 1

```
TRIGger:A:LOGIc:PATtern:INPut:CH4 {HIGH|LOW|X}
```

Syntax 2

```
TRIGger:A:LOGIc:PATtern:INPut:CH4?
```

Arguments

- `HIGH`

  This specifies the logic high.

- `LOW`

  This specifies the logic low.

- `X`

  This specifies a "do not care" state.

Example 1

```
TRIGger:A:LOGIc:PATtern:INPut:CH4?
```

This query might return `:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH`, indicating that the logic input for channel 4 is logic high.

Example 2

```
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:WHEn

Description

This command sets or queries the condition for generating an A logic pattern trigger with respect to the defined input pattern. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing a trigger condition from the Pattern drop-down list, which is located in the Trigger When group box.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:FUNCtion (see page 356), TRIGger:A:LOGIc:INPut? (see page 357), TRIGger:A:LOGIc:THReshold? (see page 377)

Syntax 1

```
TRIGger:A:LOGIc:PATtern:WHEn {TRUe|FALSe|LESSThan|MOREThan}
```

Syntax 2

```
TRIGger:A:LOGIc:PATtern:WHEn?
```

Arguments

- `TRUe`

  This specifies to trigger when the pattern becomes true.

- `FALSe`

  This specifies to trigger when the pattern becomes false.

- `LESSThan`

  This specifies to trigger if the specific pattern is true less than the specified LESSLimit.

- `MOREThan`

  This specifies to trigger if the specific pattern is true longer than the specified MORELimit.

Example 1

```
TRIGger:A:LOGIc:PATtern:WHEn?
```

This query might return `:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE`, indicating that the A logic pattern will trigger when the pattern becomes true.

Example 2

```
TRIGger:A:LOGIc:PATtern:WHEn FALSe
```

This command specifies to trigger the A logic pattern when the pattern becomes false.

TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit

Description

This command sets or queries the maximum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then entering the following Trigger When settings: choose Less Than from the Pattern drop-down list, then enter a maximum value for Time.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:PATtern:WHEn:MORELimit (see page 363)

Syntax 1

```
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit <NR3>
```

Syntax 2

```
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit?
```

Argument

- `<NR3>`

  This specifies the amount of time to hold pattern true.

Example 1

```
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit?
```

This query 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.

Example 2

```
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit 10.0E+00
```

This command 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:MORELimit

Description

This command sets or queries the minimum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then entering the following Trigger When settings: choose More Than from the Pattern drop-down list, then enter a minimum value for Time.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit (see page 362)

Syntax 1

```
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit <NR3>
```

Syntax 2

```
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit?
```

Argument

- `<NR3>`

  This specifies the amount of time to hold pattern true.

Example 1

```
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit?
```

This query 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.

Example 2

```
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit 10.0E+00
```

This command sets the minimum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

TRIGger:A:LOGIc:SETHold?

Description

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

Related Commands

TRIGger:A:LOGIc:PATtern? (see page 359), TRIGger:A:LOGIc:STATE? (see page 374)

Syntax

```
TRIGger:A:LOGIc:SETHold?
```

Example

```
TRIGger:A:LOGIc:SETHold?
```

This query might return `: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
:HOLDTIME 2.0000E-9;SETTIME 3.0000E-9
```

TRIGger:A:LOGIc:SETHold:CLOCk?

Description

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

Related Commands

TRIGger:A:LOGIc:SETHold:DATa? (see page 369), TRIGger:A:LOGIc:SETHold:CLOCk:EDGE (see page 366), TRIGger:A:LOGIc:SETHold:CLOCk:THReshold (see page 368), TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce (see page 367)

Syntax

```
TRIGger:A:LOGIc:SETHold:CLOCk?
```

Example

```
TRIGger:A:LOGIc:SETHold:CLOCk?
```

This query 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:LOGIc:SETHold:CLOCk:EDGE

Description

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

Related Commands

TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce (see page 367),
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold (see page 368)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE {FALL|RISe}
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE?
```

Arguments

- `FALL`

  This specifies polarity as the clock falling edge.

- `RISe`

  This specifies polarity as the clock rising edge.

Example 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE`, indicating that polarity is specified as the clock rising edge.

Example 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE RISE
```

This command specifies the polarity as the clock rising edge.

TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce

Description

This command sets or queries the clock source for the A logic trigger setup and hold input. The channel is specified by x, which ranges from 1 through 4. This is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired channel from the Clock Source drop-down list.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:SETHold:CLOCk:EDGE (see page 366),
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold (see page 368)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce CH<x>
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce?
```

Argument

- `CH<x>`

  This specifies the input channel, which ranges from 1 through 4.

Example 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH4`, indicating that channel 4 is the clock source for the A logic trigger's setup and hold input.

Example 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce CH1
```

This command specifies channel 1 as the A logic setup and hold input.

TRIGger:A:LOGIc:SETHold:CLOCk:THReshold

Description

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.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:SETHold:CLOCk:EDGE (see page 366), TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce (see page 367)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold {ECL|TTL|<NR3>}
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold?
```

Arguments

- `ECL`

  This specifies a preset ECL high level of -1.3 V.

- `TTL`

  This specifies a preset TTL high level of 1.4 V.

- `<NR3>`

  This is the clock level, in volts.

Example 1

```
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD 1.2000E+00`, indicating that the clock threshold for setup and hold trigger is 1.2 V.

Example 2

```
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold TTL
```

This command specifies the preset TTL value of 1.4 V as the clock threshold for setup and hold trigger.

TRIGger:A:LOGIc:SETHold:DATa?

Description

This query-only command returns the voltage threshold and data source for 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

Related Commands

TRIGger:A:LOGIc:SETHold:CLOCk? (see page 365)

Syntax

```
TRIGger:A:LOGIc:SETHold:DATa?
```

Example

```
TRIGger:A:LOGIc:SETHold:DATa?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.4000;SOURCE CH1`, indicating the current trigger data settings.

TRIGger:A:LOGIc:SETHold:DATa:THReshold

Description

This command sets or queries the data voltage threshold for setup and hold trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then setting the desired Data Level.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:SETHold:DATa:SOUrce (see page 371)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:DATa:THReshold {ECL|TTL<NR3>}
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:DATa:THReshold?
```

Arguments

- `ECL`

  This specifies the preset ECL high level of -1.3 V.

- `TTL`

  This specifies the preset TTL high level of 1.4 V.

- `<NR3>`

  This is the setup and hold data level, in V.

Example 1

```
TRIGger:A:LOGIc:SETHold:DATa:THReshold?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.2000E+00`, indicating that 1.2 V is the current data voltage level for setup and hold trigger.

Example 2

```
TRIGger:A:LOGIc:SETHold:DATa:THReshold TTL
```

This command specifies the preset ECL high level of 1.4 V as the current data voltage level for setup and hold trigger.

TRIGger:A:LOGIc:SETHold:DATa:SOUrce

Description

This command sets or queries the data source for setup and hold trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired channel from the Data Source drop-down list.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:SETHold:DATa:THReshold (see page 370)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:DATa:SOUrce CH<x>
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:DATa:SOUrce?
```

Argument

- `CH<x>`

  This specifies the input channel, which ranges from 1 through 4.

Example 1

```
TRIGger:A:LOGIc:SETHold:DATa:SOUrce?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH2`, indicating that channel 2 is the current clock source for setup and hold trigger.

Example 2

```
TRIGger:A:LOGIc:SETHold:DATa:SOUrce CH1
```

This command sets channel 1 as the clock source for setup and hold trigger.

TRIGger:A:LOGIc:SETHold:HOLDTime

Description

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

Related Commands

TRIGger:A:LOGIc:SETHold:SETTime (see page 373)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:HOLDTime <NR3>
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:HOLDTime?
```

Argument

- `<NR3>`

  This specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Example 1

```
TRIGger:A:LOGIc:SETHold:HOLDTime?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:HOLDTIME 2.0000E-09`, indicating that the current hold time for setup and hold trigger is 2 ns.

Example 2

```
TRIGger:A:LOGIc:SETHold:HOLDTime 3.0E-3
```

This command sets the hold time for A logic setup and hold trigger to 3 ms.

TRIGger:A:LOGIc:SETHold:SETTime

Description

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 Hold Time.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:SETHold:HOLDTime (see page 372)

Syntax 1

```
TRIGger:A:LOGIc:SETHold:SETTime <NR3>
```

Syntax 2

```
TRIGger:A:LOGIc:SETHold:SETTime?
```

Argument

- `<NR3>`

  This specifies the setup time for setup and hold violation triggering.

Example 1

```
TRIGger:A:LOGIc:SETHold:SETTime?
```

This query might return `:TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09`, indicating that the current hold time for setup and hold trigger is 2 ns.

Example 2

```
TRIGger:A:LOGIc:SETHold:SETTime 3.0E-6
```

This command specifies that the current hold time for setup and hold trigger is 3 µs.

TRIGger:A:LOGIc:STATE?

Description

This query-only command returns the data input and trigger criteria for the A logic trigger. This command is equivalent to selecting Logic State from the Trig menu and then viewing the current logic state settings.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:STATE:INPut:CH4 (see page 375), TRIGger:A:LOGic:STATE:WHEn (see page 376)

Syntax

```
TRIGger:A:LOGIc:STATE?
```

Example

```
TRIGger:A:LOGIc:STATE?
```

This query might return `:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE;`
`:TRIGGER:A:LOGIC:STATE:WHEN TRUE`

TRIGger:A:LOGIc:STATE:INPut:CH4

Description

This command sets or queries the slope for channel 4 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 Ch4 drop-down list.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:STATE:WHEn (see page 376)

Syntax 1

```
TRIGger:A:LOGIc:STATE:INPut:CH4 {FALL|RISe}
```

Syntax 2

```
TRIGger:A:LOGIc:STATE:INPut:CH4?
```

Arguments

- `FALL`

  This specifies falling edge.

- `RISe`

  This specifies rising edge.

Example 1

```
TRIGger:A:LOGIc:STATE:INPut:CH4?
```

This query might return `:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE`, indicating that the A logic trigger input for channel 4 is the rising edge.

Example 2

```
TRIGger:A:LOGIc:STATE:INPut:CH4 RISE
```

This command specifies that the A logic trigger input for channel 4 is the rising edge.

TRIGger:A:LOGIc:STATE:WHEn

Description

This command sets or queries the condition for generating an A logic state trigger. This command is equivalent to selecting Logic State from the Trig menu and then choosing the desired condition from the Trigger When Pattern drop-down list.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:INPut:CH<x> (see page 358), TRIGger:A:LOGIc:STATE:INPut:CH4 (see page 375)

Syntax 1

```
TRIGger:A:LOGIc:STATE:WHEn {TRUe|FALSe}
```

Syntax 2

```
TRIGger:A:LOGIc:STATE:WHEn?
```

Arguments

- `TRUe`

  This 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`

  This 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.

Example 1

```
TRIGger:A:LOGIc:STATE:WHEn?
```

This query might return `:TRIGGER:A:LOGIC:STATE:WHEN FALSE`, indicating that the logic condition to trigger upon is false.

Example 2

```
TRIGger:A:LOGIc:STATE:WHEn TRUE
```

This command specifies that the logic condition to trigger upon is true.

TRIGger:A:LOGIc:THReshold?

Description

This query-only command returns the threshold voltage for all channels in A logic trigger. This command query is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then viewing the current Input Threshold voltage settings.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:THReshold:CH<x> (see page 378), TRIGger:A:LOGIc:INPut? (see page 357), TRIGger:A:LOGIc:INPut:CH<x> (see page 358)

Syntax

```
TRIGger:A:LOGIc:THReshold?
```

Example

```
TRIGger:A:LOGIc:THReshold?
```

This query 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 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:LOGIc:THReshold:CH<x>

Description

This command sets or queries the A logic trigger threshold voltage for the channel, specified by x, which ranges from 1 through 4. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then setting the desired Input Threshold voltage for the desired channel.

Group

Trigger

Related Commands

TRIGger:A:LOGIc:INPut? (see page 357)

Syntax 1

```
TRIGger:A:LOGIc:THReshold:CH<x> <NR3>
```

Syntax 2

```
TRIGger:A:LOGic:THReshold:CH<x>?
```

Argument

- `<NR3>`

  This specifies the threshold voltage.

Example 1

```
TRIGger:A:LOGIc:THReshold:CH3?
```

This query 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.

Example 2

```
TRIGger:A:LOGIcTHReshold:CH2 3.0E-3
```

This command sets the A logic trigger threshold voltage for channel 2 to 3 mV.

TRIGger:A:MODe

Description

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

Related Commands

TRIGger:A:LEVel (see page 353)

Syntax 1

```
TRIGger:A:MODe {AUTO|NORMal}
```

Syntax 2

```
TRIGger:A:MODe?
```

Arguments

- `AUTO`

  This generates a trigger if one is not detected within a specified time period.

- `NORMal`

  This waits for a valid trigger event.

Example 1

```
TRIGger:A:MODe?
```

This query might return `:TRIGGER:A:MODE NORMAL`, indicating that a valid trigger event must occur before a trigger is generated.

Example 2

```
TRIGger:A:MODe NORMAL
```

This command specifies that a valid trigger event must occur before a trigger is generated.

TRIGger:A:PULse?

Description

This query-only command returns the A pulse trigger parameters.

Group

Trigger

Related Commands

TRIGger:A:EDGE? (see page 345), TRIGger:A:LOGIc? (see page 354)

Syntax

```
TRIGger:A:PULse?
```

Example

```
TRIGger:A:PULse?
```

This query might return

```
:TRIGGER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;POLARITY
POSITIVE;: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 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:PULse:CLAss

Description

This command sets or queries the type of pulse on which to trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired pulse Trigger Type.

Group

Trigger

Related Commands

TRIGger:A:PULse:GLItch? (see page 382), TRIGger:A:PULse:RUNT? (see page 386), TRIGger:A:PULse:WIDth? (see page 406), TRIGger:A:PULse:TIMEOut? (see page 403), TRIGger:A:PULse:TRANsition? (see page 394), TRIGger:A:TYPe (see page 412)

Syntax 1

```
TRIGger:A:PULse:CLAss {GLItch|RUNT|WIDth|TRANsition|TIMEOut}
```

Syntax 2

```
TRIGger:A:PULse:CLAss?
```

Arguments

- `GLItch`

  This triggers when a pulse is found that is of the specified polarity and width. These are set with the commands TRIGger:A:PULse:GLITch:POLarity and TRIGger:A:PULse;GLItch:WIDth.

- `RUNT`

  This triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before re-crossing the first. The thresholds are set with the TRIGger:A:PULse:RUNT:THReshold:LOW and TRIGger:A:PULse:RUNT:THReshold:HIGH commands.

- `WIDth`

  This triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by TRIGger:A;PULse:WIDth:LOWLimit and TRIGger:A:PULse:WIDth:HIGHLimit. The polarity is selected using the TRIGger:A:PULse:WIDth:POLarity command.

- `TRANsition`

  This 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`

  This triggers when the pulse train stops in the selected state for longer than the specified time.

Example 1

```
TRIGger:A:PULse:CLAss?
```

This query might return `:TRIGGER:A:PULSE:CLASS GLITCH`, indicating that a pulse was found that is of the specified polarity and width.

Example 2

```
TRIGger:A:PULse:CLAss WIDth
```

This command specifies a width pulse for the A trigger.

TRIGger:A:PULse:GLItch?

Description

This query-only command returns the current A glitch pulse trigger parameters. This command query is equivalent to selecting Glitch Setup from the Trig menu and then viewing the current pulse glitch settings.

Group

Trigger

Related Commands

TRIGger:A:PULse:GLItch:POLarity (see page 383), TRIGger:A:PULse:GLItch:TRIGIF(see page 384), TRIGger:A:PULse:GLItch:WIDth (see page 385)

Syntax

```
TRIGger:A:PULse:GLItch?
```

Example

```
TRIGger:A:PULse:GLItch?
```

This query might return `:TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09;FILTER ACCEPT;POLARITY POSITIVE`

TRIGger:A:PULse:GLItch:POLarity

Description

This command sets or queries the polarity for A pulse glitch trigger. This command is equivalent to selecting Glitch Setup from the Trig menu and then choosing the desired Polarity.

Group

Trigger

Related Commands

TRIGger:A:LEVel (see page 353), TRIGger:A:PULse:GLItch:WIDth (see page 385)

Syntax 1

```
TRIGger:A:PULse:GLItch:POLarity {POSITIVe|NEGative|EIther}
```

Syntax 2

```
TRIGger:A:PULse:GLItch:POLarity?
```

Arguments

- `POSITIVe`

  This specifies that the oscilloscope will only trigger when the polarity of the glitch is positive.

- `NEGative`

  This specifies that the oscilloscope will only trigger when the polarity of the glitch is negative.

- `EIther`

  This specifies that the oscilloscope will trigger when the polarity of the glitch is either positive or negative.

Example 1

```
TRIGger:A:PULse:GLItch:POLarity?
```

This query might return `:TRIGGER:A:PULSE:GLITCH:POLARITY POSITIVE`, indicating that the polarity of the glitch must be positive for the trigger to occur.

Example 2

```
TRIGger:A:PULse:GLItch:POLarity EITHER
```

This command specifies that the polarity of the glitch can be either positive or negative for the trigger to occur.

TRIGger:A:PULse:GLItch:TRIGIF

Description

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 then choosing the desired Trig if Width setting.

Group

Trigger

Related Commands

TRIGger:A:PULse:GLItch:WIDth (see page 385)

Syntax 1

```
TRIGger:A:PULse:GLItch:TRIGIF {ACCept|REJect}
```

Syntax 2

```
TRIGger:A:PULse:GLItch:TRIGIF?
```

Arguments

- `ACCept`

  This specifies that the oscilloscope will only trigger on pulses that are narrower than the specified width, when the trigger type is set to pulse glitch. The width is specified using the TRIGger:A:PULse:GLItch:WIDth command.

- `REJect`

  This specifies that the oscilloscope will only trigger on pulses that are wider than the specified width, when the trigger type is set to pulse glitch. The width is specified using the TRIGger:A:PULse:GLItch:WIDth command.

Example 1

```
TRIGger:A:PULse:GLItch:TRIGIF?
```

This query might return `:TRIGGER:A:PULSE:GLITCH:TRIGIF ACCEPT`, indicating that the oscilloscope is set to trigger on pulses that are narrower than the specified width.

Example 2

```
TRIGger:A:PULse:GLItch:TRIGIF REJect
```

This command specifies that the oscilloscope triggers on pulses that are wider than the specified width.

TRIGger:A:PULse:GLItch:WIDth

Description

This command sets or queries the width for the A pulse 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:PULse:GLItch:TRIGIF command.

Group

Trigger

Related Commands

TRIGger:A:LEVel (see page 353), TRIGger:A:PULse:GLItch:TRIGIF (see page 384)

Syntax 1

```
TRIGger:A:PULse:GLItch:WIDth <NR3>
```

Syntax 2

```
TRIGger:A:PULse:GLItch:WIDth?
```

Argument

- `<NR3>`

  This specifies the width of the glitch, in seconds.

Example 1

```
TRIGger:A:PULse:GLItch:WIDth?
```

This query might return `:TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09`, indicating that the width of the glitch is currently set at 2 ns.

Example 2

```
TRIGger:A:PULse:GLItch:WIDth 15E-6
```

This command sets the width of the glitch to 15 µs.

TRIGger:A:PULse:RUNT?

Description

This query-only command returns the current A runt pulse trigger parameters. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

Group

Trigger

Related Commands

TRIGger:A:PULse:GLItch? (see page 382), TRIGger:A:PULse:TIMEOut? (see page 403), TRIGger:A:PULse:TRANsition? (see page 394), TRIGger:A:PULse:WIDth? (see page 406)

Syntax

```
TRIGger:A:PULse:RUNT?
```

Example

```
TRIGger:A:PULse:RUNT?
```

This query might return `:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE;THRESHOLD:HIGH 2.0000E+00;LOW 8,0000E-01;:TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH 2.0000E-09`

TRIGger:A:PULse:RUNT:POLarity

Description

This command sets or queries the polarity for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then choosing the Polarity setting.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:THReshold? (see page 388)

Syntax 1

```
TRIGger:A:PULse:RUNT:POLarity {POSITIVe|NEGAtive|EIther}
```

Syntax 2

```
TRIGger:A:PULse:RUNT:POLarity?
```

Arguments

- `POSitive`

  This indicates that the rising edge crosses the low threshold and the falling edge re-crosses the low threshold without either edge ever crossing the high threshold.

- `NEGative`

  This indicates that the falling edge crosses the high threshold and the rising edge re-crosses the high threshold without either edge ever crossing the low threshold.

- `EIther`

  This indicates either negative or positive polarity.

Example 1

```
TRIGger:A:PULse:RUNT:POLarity?
```

This query might return `:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE`, indicating that the polarity of the A pulse runt trigger is positive.

Example 2

```
TRIGger:A:PULse:RUNT:POLarity NEGATIVE
```

This command specifies that the polarity of the A pulse runt trigger is negative.

TRIGger:A:PULse:RUNT:THReshold?

Description

This query-only command returns the upper and lower thresholds for the A pulse runt trigger. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the runt trigger Upper Level and Lower Level settings.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:POLarity (see page 387), TRIGger:A:PULse:RUNT:THReshold:BOTh (see page 389), TRIGger:A:PULse:RUNT:THReshold:HIGH (see page 390), TRIGger:A:PULse:RUNT:THReshold:LOW (see page 391)

Syntax

```
TRIGger:A:PULse:RUNT:THReshold?
```

Example

```
TRIGger:A:PULse:RUNT:THReshold?
```

This query 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:PULse:RUNT:THReshold:BOTh

Description

This command (no query form) sets the upper and lower switching thresholds for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the desired runt trigger Upper Level and Lower Level voltages.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:THReshold? (see page 388)

Syntax

```
TRIGger:A:PULse:RUNT:THReshold:BOTh {TTL|ECL}
```

Arguments

- `TTL`

  This sets the upper threshold to 1.8 V and the lower threshold to 800 mV, which are the nominal TTL voltage levels.

- `ECL`

  This sets the upper threshold to -1.1 V and the lower threshold to -1.5 V, which are the nominal ECL voltage levels.

Example

```
TRIGger:A:PULse:RUNT:THReshold:BOTh TTL
```

This command sets the threshold of the pulse runt trigger to the nominal TTL voltage levels.

TRIGger:A:PULse:RUNT:THReshold:HIGH

Description

This command sets or queries the upper limit for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the desired runt trigger Upper Level voltage.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:THReshold? (see page 388)

Syntax 1

```
TRIGger:A:PULse:RUNT:THReshold:HIGH <NR3>
```

Syntax 2

```
TRIGger:A:PULse:RUNT:THReshold:HIGH?
```

Argument

- `<NR3>`

  This specifies the threshold value, in volts.

Example 1

```
TRIGger:A:PULse:RUNT:THReshold:HIGH?
```

This query 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.

Example 2

```
TRIGger:A:PULse:RUNT:THReshold:HIGH 120E-3
```

This command sets the upper limit of the pulse runt trigger to 120 mV.

TRIGger:A:PULse:RUNT:THReshold:LOW

Description

This command sets or queries the lower limit for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the desired runt trigger Lower Level voltage.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:THReshold? (see page 388)

Syntax 1

```
TRIGger:A:PULse:RUNT:THReshold:LOW <NR3>
```

Syntax 2

```
TRIGger:A:PULse:RUNT:THReshold:LOW?
```

Argument

- `<NR3>`

  This specifies the threshold value, in volts.

Example 1

```
TRIGger:A:PULse:RUNT:THReshold:LOW?
```

This query might return `:TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW 1.2000E-01`, indicating that the lower limit of the pulse runt trigger is currently set to 120 mV.

Example 2

```
TRIGger:A:PULse:RUNT:THReshold:LOW 50E-3
```

This command sets the lower limit of the pulse runt trigger to 50 mV.

TRIGger:A:PULse:RUNT:WHEn

Description

This command sets or queries the type of pulse width the trigger checks for when it uncovers a runt. This is equivalent to selecting Runt Setup from the Trig menu and then choosing the desired Trigger When setting from the Runt drop-down list.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:WIDth (see page 393)

Syntax 1

```
TRIGger:A:PULse:RUNT:WHEn {OCCurs|WIDERthan}
```

Syntax 2

```
TRIGger:A:PULse:RUNT:WHEn?
```

Arguments

- `OCCurs`

  This specifies a trigger if a runt of any detectable width occurs.

- `WIDERthan`

  This specifies a trigger if a runt greater than the specified width occurs.

Example 1

```
TRIGger:A:PULse:RUNT:WHEn?
```

This query might return `:TRIGGER:A:PULSE:RUNT:WHEN OCCURS`, indicating that a runt trigger will occur if the oscilloscope detects a runt of any detectable width.

Example 2

```
TRIGger:A:PULse:RUNT:WHEn WIDERthan
```

This command sets the runt trigger to occur when the oscilloscope detects a runt in a pulse wider than the specified width.

TRIGger:A:PULse:RUNT:WIDth

Description

This command sets or queries the minimum width for a valid A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the desired Width.

Group

Trigger

Related Commands

TRIGger:A:PULse:RUNT:WHEn (see page 392)

Syntax 1

```
TRIGger:A:PULse:RUNT:WIDth <NR3>
```

Syntax 2

```
TRIGger:A:PULse:RUNT:WIDth?
```

Argument

- `<NR3>`

  This specifies the minimum width, in seconds.

Example 1

```
TRIGger:A:PULse:RUNT:WIDth?
```

This query might return `:TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09`, indicating that the minimum width of a pulse runt trigger is 2 ns.

Example 2

```
TRIGger:A:PULse:RUNT:WIDth 15E-6
```

This command sets the minimum width of the pulse runt trigger to 15 µs.

TRIGger:A:PULse:TRANsition?

Description

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

Related Commands

TRIGger:A:TYPe (see page 412), TRIGger:A:PULse:CLAss (see page 381), TRIGger:A:PULse:GLItch (see page 382), TRIGger:A:PULse:RUNT? (see page 386), TRIGger:A:PULse:TIMEOut? (see page 403), TRIGger:A:PULse:WIDth? (see page 406)

Syntax

```
TRIGger:A:PULse:TRANsition?
```

Example

```
TRIGger:A:PULse:TRANsition?
```

This query 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:PULse:TRANsition:DELTATime

Description

This command sets or queries the delta time used in calculating the transition value. This is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Time.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:POLarity (see page 396), TRIGger:A:PULse:TRANsition:THReshold? (see page 397)

Syntax 1

```
TRIGger:A:PULse:TRANsition:DeltaTime <NR3>
```

Syntax 2

```
TRIGger:A:PULse:TRANsition:DeltaTime?
```

Argument

- `<NR3>`

  This specifies the delta time, in seconds.

Example 1

```
TRIGger:A:PULse:TRANsition:DeltaTime?
```

This query might return `:TRIGGER:A:PULSE:TRANSITION:DELTATIME 2.0000E-09`, indicating that the transition trigger's delta time is set to 2 ns.

Example 2

```
TRIGger:A:PULse:TRANsition:DeltaTime 15E-6
```

This command sets the delta time of the transition trigger to 15 µs .

TRIGger:A:PULse:TRANsition:POLarity

Description

This command sets or queries the polarity for the A pulse transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then choosing from the Polarity pull-down list.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:DELTATime (see page 395)

Syntax 1

```
TRIGger:A:PULse:TRANsition:POLarity {POSITIVe|NEGative|EIThe r}
```

Syntax 2

```
TRIGger:A:PULse:TRANsition:POLarity?
```

Arguments

- `POSITIVe`

  This indicates that a pulse edge must traverse from the lower (most negative) to higher (post positive) level for transition triggering to occur.

- `NEGative`

  This indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

- `EIThe r`

  This indicates either positive or negative polarity.

Example 1

```
TRIGger:A:PULse:TRANsition:POLarity?
```

This query might return `:TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER` indicating that the transition can be either positive or negative.

Example 2

```
TRIGger:A:PULse:TRANsition:DeltaTime NEGative
```

This command sets the transition polarity to negative.

TRIGger:A:PULse:TRANsition:THReshold?

Description

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 then viewing the Upper Level and Lower Level voltage settings.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:DELTATime (see page 395), TRIGger:A:PULse:TRANsition:POLarity (see page 396)

Syntax

```
TRIGger:A:PULse:TRANsition:THReshold?
```

Example

```
TRIGger:A:PULse:TRANsition:THReshold?
```

This query 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:PULse:TRANsition:THReshold:BOTh

Description

This command (no query form) sets the upper and lower thresholds for pulse transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Upper Level and Lower Level voltages.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:THReshold:HIGH (see page 399),
TRIGger:A:PULse:TRANsition:THReshold:LOW (see page 400)

Syntax 1

```
TRIGger:A:PULse:TRANsition:THReshold:BOTh {TTL|ECL}
```

Arguments

- `TTL`

  This sets the upper threshold to 1.2 V and the lower threshold to 800 mV, which represent the nominal TTL voltage levels.

- `ECL`

  This sets the upper threshold to -1.1 V and the lower threshold to -1.5 V, which represent the nominal ECL voltage levels.

Example 1

```
TRIGger:A:PULse:TRANsition:THReshold:BOTh TTL
```

This command sets the thresholds of the pulse transition trigger to the nominal TTL voltage levels.

TRIGger:A:PULse:TRANsition:THReshold:HIGH

Description

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

Related Commands

TRIGger:A:PULse:TRANsition:THReshold:LOW (see page 400)

Syntax 1

```
TRIGger:A:PULse:TRANsition:THReshold:HIGH <NR3>
```

Syntax 2

```
TRIGger:A:PULse:TRANsition:THReshold:HIGH?
```

Argument

- `<NR3>`

  This specifies the upper threshold, in volts.

Example 1

```
TRIGger:A:PULse:TRANsition:THReshold:HIGH?
```

This query might return `:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 2.0000E+00`, indicating that the upper limit of the pulse transition trigger is 2 V.

Example 2

```
TRIGger:A:PULse:TRANsition:THReshold:HIGH 120E-3
```

This command sets the upper limit of the pulse transition trigger to 120 mV.

TRIGger:A:PULse:TRANsition:THReshold:LOW

Description

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 then setting the desired Lower Level voltage.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:THReshold:HIGH (see page 399)

Syntax 1

```
TRIGger:A:PULse:TRANsition:THReshold:LOW <NR3>
```

Syntax 2

```
TRIGger:A:PULse:TRANsition:THReshold:LOW?
```

Argument

- `<NR3>`

  This specifies the lower threshold, in volts.

Example 1

```
TRIGger:A:PULse:TRANsition:THReshold:LOW?
```

This query might return `:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW 50.0000E-03`, indicating that the lower limit of the pulse transition trigger is 50 mV.

Example 2

```
TRIGger:A:PULse:TRANsition:THReshold:LOW 20E-3
```

This command sets the lower limit of the pulse transition trigger to 20 mV.

TRIGger:A:PULse:TRANsition:WHEn

Description

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 then choosing the desired Trigger When Transition Time setting.

Group

Trigger

Related Commands

TRIGger:A:PULse:TRANsition:DELTATime (see page 395), TRIGger:A:PULse:TRANsition:POLarity (see page 396), TRIGger:A:PULse:TRANsition:THReshold? (see page 397)

Syntax 1

```
TRIGger:A:PULse:TRANsition:WHEn {FASTERthan|SLOWERthan}
```

Syntax 2

```
TRIGger:A:PULse:TRANsition:WHEn?
```

Arguments

- `FASTERthan`

  This sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

- `SLOWERthan`

  This sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

Example 1

```
TRIGger:A:PULse:TRANsition:WHEn?
```

This query 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.

Example 2

```
TRIGger:A:PULse:TRANsition:WHEn SLOWERthan
```

This command sets the transition trigger to work when the transitioning signal is slower than the set volts/second rate.

TRIGger:A:PULse:SOUrce

Description

This command sets or queries the source for the A pulse trigger. This source parameter applies to all classes of pulse trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then choosing the desired channel from the Source pull-down list.

Group

Trigger

Related Commands

TRIGger:A:EDGE:SOUrce (see page 348)

Syntax 1

```
TRIGger:A:PULse:SOUrce CH<x>
```

Syntax 2

```
TRIGger:A:PULse:SOUrce?
```

Argument

- `CH<x>`

  This specifies one of the input channels, which range from 1 through 4.

Example 1

```
TRIGger:A:PULse:SOUrce?
```

This query might return `:TRIGGER:A:PULSE:SOURCE CH2`, indicating that channel 2 is the source for the A pulse trigger.

Example 2

```
TRIGger:A:PULse:SOUrce CH4
```

This command sets channel 4 as the source for the A pulse trigger.

TRIGger:A:PULse:TIMEOut?

Description

This query-only command returns the polarity and time-out duration for the A pulse time-out trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then viewing the current Polarity and Time settings.

Group

Trigger

Related Commands

TRIGger:A:PULse:TIMEOut:POLarity (see page 404), TRIGger:A:PULse:TIMEOut:TIMe (see page 405)

Syntax

```
TRIGger:A:PULse:TIMEOut?
```

Example

```
TRIGger:A:PULse:TIMEOut?
```

This query might return `:TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYSHIGH;TIME 2.0000E-9`, indicating that the A current polarity setting for the pulse trigger is STAYSHIGH (positive) and the current timeout duration is 2 ns.

TRIGger:A:PULse:TIMEOut:POLarity

Description

This command sets or queries the polarity for the A pulse time-out trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Polarity.

Group

Trigger

Related Commands

TRIGger:A:PULse:TIMEOut? (see page 403)

Syntax 1

```
TRIGger:A:PULse:TIMEOut:POLarity {STAYSHigh|STAYSLow|EITher}
```

Syntax 2

```
TRIGger:A:PULse:TIMEOut:POLarity?
```

Arguments

- `STAYSHigh`

  This 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`

  This indicates that a pulse edge must stay low (negative) for the required time period to permit time-out triggering to occur.

- `EITher`

  This 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.

Example 1

```
TRIGger:A:PULse:TIMEOut:POLarity?
```

This query might return `:TRIGGER:A:PULSE:TIMEOUT:POLARITY EITHER`, indicating that the polarity of the A pulse time-out trigger can be either positive or negative.

Example 2

```
TRIGger:A:PULse:TIMEOut:POLarity STAYSHigh
```

This command sets the polarity of the A pulse time-out trigger to positive.

TRIGger:A:PULse:TIMEOut:TIMe

Description

This command sets or queries the pulse time-out trigger time, in seconds. This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired time-out Time.

Group

Trigger

Related Commands

TRIGger:A:PULse:TIMEOut? (see page 403)

Syntax 1

```
TRIGger:A:PULse:TIMEOut:TIMe <NR3>
```

Syntax 2

```
TRIGger:A:PULse:TIMEOut:TIMe?
```

Argument

- `<NR3>`

  This specifies the time-out time period, in seconds.

Example 1

```
TRIGger:A:PULse:TIMEOut:TIMe?
```

This query might return `:TRIGGER:A:PULSE:TIMEOUT:TIME 2.0000E-9`, indicating that the time-out time is currently set to 2 ns.

Example 2

```
TRIGger:A:PULse:TIMEOut:TIMe 3.134E-6
```

This command sets the time-out time to 3.134 µs.

TRIGger:A:PULse:WIDth?

Description

This query-only command returns the width parameters for the A 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

Related Commands

TRIGger:A:PULse:WIDth:HIGHLimit (see page 407), TRIGger:A:PULse:WIDth:LOWLimit (see page 408), TRIGger:A:PULse:WIDth:POLarity (see page 409), TRIGger:A:PULse:WIDth:WHEn (see page 410)

Syntax 1

```
TRIGger:A:PULse:WIDth
```

Example

```
TRIGger:A:PULse:WIDth?
```

This query might return `:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;POLARITY POSITIVE` as the current A pulse trigger parameters.

TRIGger:A:PULse:WIDth:HIGHLimit

Description

This command sets or queries the upper limit for the A pulse width trigger. This command is equivalent to selecting Width Setup from the Trig menu and then setting the pulse width trigger Upper Limit.

Group

Trigger

Related Commands

TRIGger:A:PULse:WIDth? (see page 406)

Syntax 1

```
TRIGger:A:PULse:WIDth:HIGHLimit <NR3>
```

Syntax 2

```
TRIGger:A:PULse:WIDth:HIGHLimit?
```

Argument

- `<NR3>`

  This specifies the A pulse width trigger upper limit, in seconds.

Example 1

```
TRIGger:A:PULse:WIDth:HIGHLimit?
```

This query might return `:TRIGGER:A:PULSE:WIDTH:HIGHLIMIT 2.0000E-9`, indicating that the A pulse upper limit is set to 2 ns.

Example 2

```
TRIGger:A:PULse:WIDth:HIGHLimit 5.0E-6
```

This command sets the A pulse upper limit to 5 µs.

TRIGger:A:PULse:WIDth:LOWLimit

Description

This command sets or queries the lower limit for the A pulse width trigger. This command is equivalent to selecting Width Setup from the Trig menu and then setting the pulse width trigger Lower Limit.

Group

Trigger

Related Commands

TRIGger:A:PULse:WIDth? (see page 393)

Syntax 1

```
TRIGger:A:PULse:WIDth:LOWLimit <NR3>
```

Syntax 2

```
TRIGger:A:PULse:WIDth:LOWLimit?
```

Argument

- `<NR3>`

  This specifies the A pulse width trigger lower limit, in seconds.

Example 1

```
TRIGger:A:PULse:WIDth:LOWLimit?
```

This query might return `:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 1.0000E-9`, indicating that the A pulse lower limit is set to 1 ns.

Example 2

```
TRIGger:A:PULse:WIDth:LOWLimit 2.0E-6
```

This command sets the A pulse lower limit to 2 µs.

TRIGger:A:PULse:WIDth:POLarity

Description

This command sets or queries the polarity for the A pulse width trigger. This command is equivalent to selecting Width Setup from the Trig menu and then selecting the pulse width trigger Polarity.

Group

Trigger

Related Commands

TRIGger:A:PULse:WIDth? (see page 406)

Syntax 1

```
TRIGger:A:PULse:WIDth:POLarity {NEGAtive|POSITIVe}
```

Syntax 2

```
TRIGger:A:PULse:WIDth:POLarity?
```

Arguments

- `NEGAtive`

  This specifies a negative pulse.

- `POSITIVe`

  This specifies a positive pulse.

Example 1

```
TRIGger:A:PULse:WIDth:POLarity?
```

This query might return `:TRIGGER:A:PULSE:WIDTH:POLARITY POSITIVE,` indicating a positive pulse.

Example 2

```
TRIGger:A:PULse:WIDth:POLarity NEGAtive
```

This command sets the pulse polarity to negative.

TRIGger:A:PULse:WIDth:WHEn

Description

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

Related Commands

TRIGger:A:PULse:WIDth:HIGHLimit (see page 407), TRIGger:A:PULse:WIDth:LOWLimit (see page 408)

Syntax 1

```
TRIGger:A:PULse:WIDth:WHEn {OUTside|WIThin}
```

Syntax 2

```
TRIGger:A:PULse:WIDth:WHEn?
```

Arguments

- `OUTside`

  This specifies a trigger when 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`

  This specifies a trigger 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.

Example 1

```
TRIGger:A:PULse:WIDth:WHEn?
```

This query might return `:TRIGGER:A:PULSE:WIDTH:WHEN OUTSIDE`, indicating the conditions for generating a pulse trigger.

Example 2

```
TRIGger:A:PULse:WIDth:WHEn WIThin
```

This command specifies that the duration of the A pulse will fall within defined high and low limits.

TRIGger:STATE?

Description

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

Related Commands

TRIGger:A:MODe (see page 379)

Syntax

```
TRIGger:STATE?
```

Outputs

- `ARMed`

  This indicates that the oscilloscope is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is arming.

- `AUTO`

  This indicates that the oscilloscope is in the auto mode and acquires data even in the absence of a trigger.

- `DPO`

  This indicates that the oscilloscope is in DPO mode.

- `PARTial`

  This indicates that the A trigger has occurred and the oscilloscope is waiting for the B trigger to occur.

- `REAdy`

  This indicates that all pretrigger information has been acquired and that the oscilloscope is ready to accept a trigger.

- `SAVe`

  This indicates that the oscilloscope is in save mode and is not acquiring data.

- `TRIGger`

  This indicates that the oscilloscope has seen a trigger and is acquiring the post-trigger information.

Example

```
TRIGger:STATE?
```

This query might return `:TRIGGER:STATE ARMED`, indicating that the pre-trigger data is being acquired.

TRIGger:A:TYPe

Description

This command sets or queries the type of A trigger. The three types of triggers consist of Edge, Logic, and Pulse. Both Logic and Pulse triggers contain classes. Logic triggers consist of State and Pattern classes; Pulse triggers consist of Glitch, Runt, Width, Transition, and Timeout 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:LOGIc:CLAss and TRIGger:A:PULse:CLAss respectively. This command is similar to selecting A Event (Main) Trigger Setup from the Trig menu and then selecting the desired Trigger Type.

Group

Trigger

Related Commands

TRIGger:A:EDGE? (see page 345), TRIGger:A:LOGIc:CLAss (see page 355), TRIGger:A:PULse:CLAss (see page 381)

Syntax 1

```
TRIGger:A:TYPe {EDGE|LOGIc|PULse}
```

Syntax 2

```
TRIGger:A:TYPe?
```

Arguments

- `EDGE`

  This 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`

  This specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:A:LOGIc commands.

- `PULse`

  This specifies that a trigger occurs when a specified pulse is found and is controlled by the TRIGger:A:PULse commands.

Example 1

```
TRIGger:A:TYPe?
```

This query might return `:TRIGGER:A:TYPE PULSE,` indicating that the A trigger type is a pulse trigger.

Example 2

```
TRIGger:A:TYPe EDGE
```

This command sets the A trigger type to EDGE.

TRIGger:B

Description

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

Related Commands

TRIGger:A (see page 344)

Syntax 1

```
TRIGger:B SETLevel
```

Syntax 2

```
TRIGger:B?
```

Argument

- `SETLevel`

  This sets the B trigger level to 50% of MIN and MAX.

Example 1

```
TRIGger:B?
```

This query might return `: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`

Example 2

```
TRIGger:B SETLevel
```

This command sets the B trigger level to 50% of MIN and MAX.

TRIGger:B:BY

Description

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

Related Commands

TRIGger:B:EVENTS:COUNt (see page 420), TRIGger:B:TIMe (see page 423), HORIZontal[:MAIN]:DELay:MODe (see page 184), HORIZontal[:MAIN]:DELay:TIMe (see page 185)

Syntax 1

```
TRIGger:B:BY {EVENTS|TIMe}
```

Syntax 2

```
TRIGger:B:BY?
```

Arguments

- `EVENTS`

  This 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`

  This sets the B trigger to be ready to occur a set time after the A trigger event. The time period is specified by TRIGger:B:TIMe.

Example 1

```
TRIGger:B:BY?
```

This query 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.

Example 2

```
TRIGger:B:BY TIMe
```

This command sets the B trigger to be ready to occur a set time after the A trigger event.

TRIGger:B:EDGE?

Description

This query-only command returns the source, slope, and coupling for the edge type of B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and then viewing the current Source, Slope, and Coupling settings.

Group

Trigger

Related Commands

TRIGger:B:EDGE:COUPling (see page 416), TRIGger:B:EDGE:SLOpe (see page 417), TRIGger:B:EDGE:SOUrce (see page 418)

Syntax

```
TRIGger:B:EDGE?
```

Example

```
TRIGger:B:EDGE?
```

This query might return `:TRIGGER:B:EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC`

TRIGger:B:EDGE:COUPling

Description

This command sets or queries the type of coupling for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and then choosing the desired setting from the Coupling drop-down list.

Group

Trigger

Related Commands

TRIGger:B:EDGE? (see page 415)

Syntax 1

```
TRIGger:B:EDGE:COUPling {DC|ATRIGger|NOISErej}
```

Syntax 2

```
TRIGger:B:EDGE:COUPling?
```

Arguments

- `ATRIGger`

  This sets the B trigger coupling to match the setting on the A trigger.

- `DC`

  This selects DC trigger coupling.

- `NOISErej`

  This selects DC low sensitivity.

Example 1

```
TRIGger:B:EDGE:COUPling?
```

This query might return `:TRIGGER:B:EDGE:COUPLING ATRIGGER` for the B trigger coupling.

Example 2

```
TRIGger:B:EDGE:COUPling DC
```

This command selects DC for the B trigger coupling.

TRIGger:B:EDGE:SLOpe

Description

This command sets or queries the slope for the B edge trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and then choosing the desired Slope.

Group

Trigger

Related Commands

TRIGger:B:EDGE? (see page 415)

Syntax 1

```
TRIGger:B:EDGE:SLOpe {RISe|FALL}
```

Syntax 2

```
TRIGger:B:EDGE:SLOpe?
```

Arguments

- `RISe`

  This specifies the trigger on the rising or positive edge of a signal.

- `FALL`

  This specifies the trigger in the falling or negative edge of a signal.

Example 1

```
TRIGger:B:EDGE:SLOpe?
```

This query might return `:TRIGGER:B:EDGE:SLOPE RISE`, indicating that the B edge trigger occurs on the rising slope.

Example 2

```
TRIGger:B:EDGE:SLOpe FALL
```

This command sets the B edge trigger to occur on the falling slope.

TRIGger:B:EDGE:SOUrce

Description

This command sets or queries the source for the B edge trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and then choosing the desired setting from the Source drop-down list.

Group

Trigger

Related Commands

TRIGger:B:EDGE? (see page 415)

Syntax 1

```
TRIGger:B:EDGE:SOUrce {AUXiliary|CH<x>}
```

Syntax 2

```
TRIGger:B:EDGE:SOUrce?
```

Arguments

- `AUXiliary`

  This specifies an external trigger (using the Auxiliary Trigger Input connector, located on the rear panel of the oscilloscope) as the B trigger source.

- `CH<x>`

  This specifies one of the input channels as the delay trigger source. Input channels are specified by x, which can range from 1 through 4.

Example 1

```
TRIGger:B:EDGE:SOUrce?
```

This query might return `:TRIGGER:B:EDGE:SOURCE CH1`, indicating that the current input source for the B trigger is channel 1.

Example 2

```
TRIGger:B:EDGE:SOUrce CH4
```

This command sets channel 4 as the input source for the B trigger.

TRIGger:B:EVENTS?

Description

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 then viewing the Trig Event setting.

Group

Trigger

Related Commands

TRIGger:B:EVENTS:COUNt (see page 420)

Syntax

```
TRIGger:B:EVENTS?
```

Example

```
TRIGger:B:EVENTS?
```

This query might return `:TRIGGER:B:EVENTS:COUNT 2`, indicating that 2 events must occur before the B trigger occurs.

TRIGger:B:EVENTS:COUNt

Description

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 then setting the desired Trig Event value.

Group

Trigger

Related Commands

TRIGger:B:EVENTS? (see page 419)

Syntax 1

```
TRIGger:B:EVENTS:COUNt <NR1>
```

Syntax 2

```
TRIGger:B:EVENTS:COUNt?
```

Argument

- `<NR1>`

  This is the number of B edge trigger events, which can range from 1 to 10,000,000.

Example 1

```
TRIGger:B:EVENTS:COUNt?
```

This query might return `:TRIGGER:B:EVENTS:COUNT 2`, indicating that two events must occur after the A trigger before the B trigger can occur.

Example 2

```
TRIGger:B:EVENTS:COUNt 4
```

This command specifies that the B trigger will occur four trigger events after the A trigger.

TRIGger:B:LEVel

Description

This command sets or queries the level for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A→B Seq tab, and then setting the B Trig Level voltage.

Group

Trigger

Related Commands

TRIGger:A:LEVEL (see page 353), TRIGger:B (see page 413), TRIGger:B:EDGE:SOUrce (see page 418)

Syntax 1

```
TRIGger:B:LEVel {ECL|TTL|<NR3>}
```

Syntax 2

```
TRIGger:B:LEVel?
```

Arguments

- `ECL`

  This specifies a preset ECL level of -1.3 V.

- `TTL`

  This specifies a preset TTL level of 1.4 V.

- `<NR3>`

  This is the B trigger level, in volts.

Example 1

```
TRIGger:B:LEVel?
```

This query might return `:TRIGGER:B:LEVEL 173.0000E-03`, indicating that the B trigger level is currently set at 173 mV.

Example 2

```
TRIGger:B:LEVel ECL
```

This command sets the B trigger level to -1.3 V.

This command sets the B trigger level to 1.4 V.

TRIGger:B:STATE

Description

This command sets or queries the state of B trigger activity. If the trigger B state is on, the B trigger is part of the triggering sequence. If trigger B state is off, then only the A trigger causes the trigger event.

Group

Trigger

Related Commands

TRIGger:A:MODe (see page 379)

Syntax 1

```
TRIGger:B:STATE {ON|OFF|<NR1>}
```

Syntax 2

```
TRIGger:B:STATE?
```

Arguments

- `ON`

  This indicates that the B trigger is active and in conjunction with the A trigger causes trigger events.

- `OFF`

  This indicates that only the A trigger causes trigger events.

- `<NR1>`

  A 0 turns off the B trigger; any other value activates the B trigger.

Example 1

```
TRIGger:B:STATE?
```

This query might return `:TRIGGER:B:STATE 0`, indicating that the B trigger is inactive and that only the A trigger causes trigger events.

Example 2

```
TRIGger:B:STATE ON
```

This command sets the B trigger to active, making it capable of causing trigger events.

TRIGger:B:TIMe

Description

This command sets or queries trigger B delay time. Trigger B 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 then setting Trig Delay.

Group

Trigger

Related Commands

TRIGger:B:BY (see page 414), TRIGger:B:EVENTS:COUNt (see page 420)

Syntax 1

```
TRIGger:B:TIMe <NR3>
```

Syntax 2

```
TRIGger:B:TIMe?
```

Argument

- `<NR3>`

  This is the B trigger time, in seconds.

Example 1

```
TRIGger:B:TIMe?
```

This query might return `:TRIGGER:B:TIME 16.0000E-9`, indicating that trigger B time is currently set to 16 ns.

Example 2

```
TRIGger:B:TIMe 4E-6
```

This command sets trigger B time to 4 µs.

TRIGger:B:TYPe

Description

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 then choosing Edge.

Group

Trigger

Related Commands

TRIGger:A:TYPe (see page 412)

Syntax 1

```
TRIGger:B:TYPe EDGE
```

Syntax 2

```
TRIGger:B:TYPe?
```

Argument

- `EDGE`

  This sets the B trigger type to edge.

Example 1

```
TRIGger:B:TYPe?
```

This query will return `:TRIGGER:B:TYPE EDGE`

Example 2

```
TRIGger:B:TYPe EDGE
```

This command sets the B trigger type to edge.

Vertical Command Group

Vertical Overview

You 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. This group contains commands to set up vertical parameters for electrical or optical signals, depending on the sampling module in use.

The `SELect:<wfm>` command also selects the waveform that many commands in other command groups use.

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

Vertical Commands

Command	Description
CH<x>?	Returns vertical parameters for the specified channel
CH<x>:BANdwidth?	Returns the bandwidth of the specified channel
CH<x>:BANdwidth	Sets the bandwidth of the specified channel
CH<x>:COUPling?	Returns the coupling setting for the specified channel
CH<x>:COUPling	Sets the coupling for the specified channel
CH<x>:DESKew?	Returns the deskew time for the specified channel
CH<x>:DESKew	Sets the deskew time for the specified channel
CH<x>:LABEL:NAMe?	Returns the label defined for the channel waveform
CH<x>:LABEL:NAMe	Defines the label for the channel waveform
CH<x>:LABEL:XPOS?	Returns the X display coordinate for the channel waveform label
CH<x>:LABEL:XPOS	Sets the X display coordinate for the channel waveform label
CH<x>:LABEL:YPOS?	Returns the Y display coordinate for the channel waveform label
CH<x>:LABEL:YPOS	Sets the Y display coordinate for the channel waveform label
CH<x>:OFFSet?	Returns the channel offset setting
CH<x>:OFFSet	Sets the channel offset
CH<x>:POSition?	Returns the channel vertical position
CH<x>:POSition	Sets 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>:PROBECal?	Returns the probe calibration status
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: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>:PROBEFunc:EXTatten?	Returns the user-specified attenuation
CH<x>:PROBEFunc:EXTatten	Sets the attenuation value for the specified channel to the specified scale factor
CH<x>PROBEFunc:EXTDBatten ?	Returns the user-specified attenuation, in decibels
CH<x>PROBEFunc:EXTDBatten	Sets the attenuation value for the specified channel to the specified value, in decibels
CH<x>PROBEFunc:EXTUnits?	Returns the user-specified unit of measurement for the external attenuator
CH<x>PROBEFunc:EXTUnits	Sets the unit of measurement for the external attenuator of the specified channel
CH<x>:SCAle?	Returns the vertical scale (per div) of the specified channel
CH<x>:SCAle	Sets the vertical scale (per div) of the specified channel
CH<x>:TERmination?	Returns channel input termination
CH<x>:TERmination	Sets channel input termination
MATH<x>:POSition?	Returns the vertical position for math display
MATH<x>:POSition	Sets the vertical position for math display
MATH<x>:SCAle?	Returns the vertical scale (per div) for math display
MATH<x>:SCAle	Sets the vertical scale (per div) for math display
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?	Returns the label of the designated waveform
REF<x>:LABel:NAMe	Sets the label of the designated waveform
REF<x>:LABel:XPOS?	Returns the position of the reference waveform label on the X axis
REF<x>:LABel:XPOS	Sets the position of the reference waveform label on the X axis
REF<x>:LABel:YPOS?	Returns the position of the reference waveform label on the Y axis
REF<x>:LABel:YPOS	Sets the position of the reference waveform label on the Y axis
REF<x>:POSition?	Returns the vertical position of the specified reference waveform
REF<x>:POSition	Sets the vertical position of the specified reference waveform
REF<x>:SCAle?	Returns the vertical scale of the

	specified reference waveform
REF<x>:SCAle	Sets the vertical scale of the specified reference waveform
SELect?	Return information on which waveforms are ON/OFF and which is the selected waveform.
SELect:<wfm>?	Returns whether the specified channel is on or off
SELect<wfm>	Turns on the specified waveform
SELect:CONTRol?	Returns the waveform and time base selected for front-panel control.
SELect:CONTRol <wfm>	Sets the waveform controlled by front panel.

CH<x>?

Description

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x, which ranges from 1 through 4. This command is similar to selecting Vertical Setup from the Vertical menu.

Group

Vertical

Syntax

```
CH<x>?
```

Example

```
CH1?
```

This query might return :CH1:BANDWIDTH 1.0000E+09; COUPLING DC;DESKEW 0.0000E+00;OFFSET 0.0000E+00;POSITION 0.0000E+00 ;SCALE 5.0000E-01;TERMINATION 1.0000E+06;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, showing the vertical parameters for channel 1.

CH<x>:BANdwidth

Description

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. This is equivalent to selecting Bandwidth from the Vertical menu.

Group

Vertical

Syntax 1

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

Syntax 2

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

Arguments

- `TWEnty`

  This sets the upper bandwidth limit to 20 MHz.

- `TWOfifty`

  This sets the upper bandwidth limit to 250 MHz.

- `FULl`

  This disables any optional bandwidth limiting. The specified channel operates at its maximum attainable bandwidth.

- `<NR3>`

  This represents a double-precision ASCII string, which is then rounded to available bandwidths using geometric rounding and sets the upper bandwidth.

Example 1

```
CH1:BANdwidth TWEnty
```

This command sets the bandwidth of channel 1 to 20 MHz.

Example 2

```
CH2:BANdwidth?
```

This query might return `:CH2:BANDWIDTH 1.0000E+09`, indicating that there is no bandwidth limiting on channel 2.

CH<x>:COUPling

Description

This command sets or queries the input attenuator coupling setting for the specified channel. This command is equivalent to selecting Coupling from the Vertical menu.

Group

Vertical

Syntax 1

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

Syntax 2

```
CH<x>:COUPling?
```

Arguments

- `AC`

  This sets the specified channel to AC coupling.

- `DC`

  This sets the specified channel to DC coupling.

- `GND`

  This sets the specified channel to ground. Only a flat, ground-level waveform will be displayed.

Example 1

```
CH2:COUPling AC
```

This command sets channel 2 to AC coupling.

Example 2

```
CH3:COUPling?
```

This query might return `:CH3:COUPling DC`, indicating that channel 3 is set to DC coupling.

CH<x>:DESKew

Description

This command sets or queries the deskew time for the specified channel. The channel is specified by x, which 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 1

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

Syntax 2

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

Argument

- `<NR3>`

  This 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.

Example 1

```
CH4:DESKew 5.0E-9
```

This command sets the deskew time for channel 4 to 5 ns.

Example 2

```
CH2:DESKew?
```

This query might return `:CH2:DESKEW 2.0000E-09`, indicating that the deskew time for channel 2 is set to 2 ns.

CH<x>:LABEL:NAMe

Description

This command sets or queries the label attached to the displayed waveform for the specified channel. This command is equivalent to selecting Label from the Vertical menu.

Group

Vertical

Syntax 1

```
CH<x>:LABEL:NAMe <str>
```

Syntax 2

```
CH<x>:LABEL:NAMe?
```

Argument

- `<str>`

  This is an alphanumeric character string, ranging from 1 through 32 characters in length.

Example 1

```
CH2:LABEL:NAMe "Pressure"
```

This command changes the waveform label for the CH2 waveform to "Pressure".

Example 2

```
CH3:LABEL:NAMe?
```

This query might return `:CH3:LABEL:NAME "CH3"`, indicating that the waveform label for the CH 3 waveform is "CH3".

CH<x>:LABEL:XPOS

Description

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. Channels are specified by x, which ranges 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 1

```
CH<x>:LABEL:XPOS <NR1>
```

Syntax 2

```
CH<x>:LABEL:XPOS?
```

Argument

- `<NR1>`

  This 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.

Example 1

```
CH3:LABEL:XPOS 50
```

This command moves the waveform label for the CH3 waveform, so that it begins 50 pixels to the right of the left edge of the screen.

Example 2

```
CH2:LABEL:XPOS?
```

This query 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

Description

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. Channels are specified by x, which ranges 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 1

```
CH<x>:LABEL:YPOS <NR1>
```

Syntax 2

```
CH<x>:LABEL:YPOS?
```

Argument

- `<NR1>`

  This 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 400.

Example 1

```
CH3:LABEL:YPOS -25
```

This command moves the waveform label for the CH3 waveform to just beneath (25 pixels below) the top of the screen.

Example 2

```
CH2:LABEL:YPOS?
```

This query 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

Description

This command sets or queries the vertical offset for the specified channel. The channel is specified by x, which 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.

The settable range of a channel offset is either ±100 V, ±10V or ±1.0 V, depending on the vertical scale factor.

Vertical Scale Adjust Factors			
When Internal scale adjust gain is:	From 0.001 to 0.1	From 0.101 to 01.0	From 1.01 to 100.0
Product of Offset range, probe gain and transducer gain is:	±1.0 V	±10.0 V	±100.0 V

Note: The above table describes oscilloscope behavior only when no probe is attached and when the external attenuation factor is 1.0.

Group

Vertical

Related Commands

CH<x>POSition (see page 437)

Syntax 1

```
CH<x>:OFFSet <NR3>
```

Syntax 2

```
CH<x>:OFFSet?
```

Argument

- `<NR3>`

  This is the offset value for the specified channel, ranging from -1.6V through 1.6V.

Example 1

```
CH3:OFFSet 2.0E-3
```

This command sets the offset for channel 3 to 2mV.

Example 2

```
CH2:OFFSet?
```

This query might return `:CH4:OFFSET 1.0000E-03`, indicating that the offset for channel 4 is set to 1 mV.

CH<x>:POSition

Description

This command sets or queries the vertical position of the specified channel. The channel is specified by x, which ranges 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 channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal 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

Related Commands

CH<x>:OFFSet (see page 436), REF<x>:POSition (see page 457), MATH<x>:POSition (see page 206)

Syntax 1

```
CH<x>:POSition <NR3>
```

Syntax 2

```
CH<x>:POSition?
```

Argument

- `<NR3>`

  This is the position value, in divisions from the center graticule, ranging from 8 to -8 divisions.

Example 1

```
CH2:POSition 1.3E+00
```

This command positions the Channel 2 input signal 1.3 divisions above the center graticule.

Example 2

```
CH1:POSition?
```

This query 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?

Description

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x, which ranges from 1 through 4.

Group

Vertical

Related Commands

CALibrate:PRObestate:CH<x>? (see page 43), CH<x>:PROBECal? (see page 439)

Syntax

```
CH<x>:PROBE?
```

Example

```
CH2:PROBE?
```

This query 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 the attenuation factor for the probe attached to channel 2 is 100.0 mV (assuming that probe units are set to volts).

CH<x>:PROBECal?

Description

This query-only command returns the probe calibration state for the selected channel. The channel is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Probe Cal from the Vertical menu.

Group

Vertical

Related Commands

CALibrate:PRObestate:CH<x>? (see page 43)

Syntax

```
CH<x>:PROBECal?
```

Outputs

- `FAIl`

  This signifies that the probe calibration has failed for the selected channel.

- `INIT`

  This signifies that the probe calibration has not yet been run for the selected channel.

- `PASS`

  This signifies the probe calibration has passed for the selected channel.

Example

```
CH2:PROBECal?
```

This query might return `:CH2:PROBECAL PASS`, indicating that the probe calibration has passed for channel 2.

CH<x>:PRObe:GAIN?

Description

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x, which ranges 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

Related Commands

CH<x>:SCAle (see page 449)

Syntax

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

Example

```
CH2:PRObe:GAIN?
```

This query 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 10 V applied to the probe input.

CH<x>:PRObe:ID?

Description

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, which ranges from 1 through 4.

Group

Vertical

Syntax

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

Example

```
CH2:PRObe:ID?
```

This query 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:TYPe?

Description

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x, which ranges 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?
```

Example

```
CH1:PRObe:ID:TYPe?
```

This query might return `:CH1:PROBE:ID:TYPE "P6203"`, indicating that P6203-type probe is attached to channel 1.

CH<x>:PRObe:ID:SERnumber?

Description

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x, which ranges 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?
```

Example

```
CH1:PRObe:ID:SERnumber?
```

This query might return `:CH1:PROBE:ID:SERNUMBER "B010289"`, indicating that the serial number of the probe attached to channel 1 is B010289.

CH<x>:PRObe:RESistance?

Description

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

Group

Vertical

Syntax

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

Example

```
CH2:PRObe:RESistance?
```

This query might return `:CH2:PROBE:RESISTANCE 10.0000E+06`, indicating that the input resistance of the probe attached to Channel 2 is 10.0 megOhms.

CH<x>:PRObe:UNIts?

Description

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, which ranges from 1 through 4.

Group

Vertical

Related Commands

CH<x>:PROBEFunc:EXTUnits (see page 448)

Syntax

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

Example

```
CH4:PRObe:UNIts?
```

This query might return `:CH4:PROBE:UNITS "V"`, indicating that the units of measure for the probe attached to channel 4 are volts.

CH<x>:PROBEFunc:EXTatten

Description

This command instructs the oscilloscope when to make vertical settings (offset and scale) for the specified channel effective. The vertical settings become effective at the input of the external network or transducer (of a specified input÷output transfer ratio) when the output is connected to the input of the probe attached to the channel.

The channel is specified by x, which ranges from 1 through 4. There is also a corresponding query, which returns the user-specified attenuation. Note that, as the name implies, this command deals with an attenuation factor, not a gain factor, unlike CH<x>:PROBe? (Note that this command returns a value independent of the External Attenuation). For example, if you specify a 20x attenuation factor for channel 1, then 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 command and then viewing or setting Ext Atten.

Group

Vertical

Related Commands

CH<x>:PROBEFunc:EXTDBatten (see page 447)

Syntax 1

```
CH<x>:PROBEFunc:EXTatten <NR3>
```

Syntax 2

```
CH<x>:PROBEFunc:EXTatten?
```

Argument

- `<NR3>`

  This is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

Example 1

```
CH1:PROBEFunc:EXTatten 167.00E-3
```

This command specifies an external 'gain of six' network, which is connected between the user's input signal and the input of the probe attached to channel 1.

Example 2

```
CH2:PROBEFunc:EXTatten?
```

This query 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

Description

This command instructs the oscilloscope to make the vertical scale and offset settings for a specified channel effective at the input of an external network or transducer (with a specified input ÷ output transfer ratio in decibels). It is assumed that the input of the probe for the specified channel is connected to the output of this network or transducer. The channel is specified by x, which ranges from 1 through 4

There is also a corresponding query, which returns the user-specified attenuation, in decibels. Note that 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

Related Commands

CH<x>:PROBEFunc:EXTatten (see page 446)

Syntax 1

```
CH<x>:PROBEFunc:EXTDBatten <NR3>
```

Syntax 2

```
CH<x>:PROBEFunc:EXTDBatten?
```

Argument

- `<NR3>`

  This is the attenuation value, which is specified in the range from -200.00 to 200.00.dB.

Example 1

```
CH3:PROBEFunc:EXTDBatten 2.5
```

This command specifies an external 2.5 dB attenuator on channel 3.

Example 2

```
CH1:PROBEFunc:EXTDBatten?
```

This query might return `:CH1:PROBEFUNC:EXTDBATTEEN 2.5000E+00`, indicating that the attenuation for channel 1 is 2.5 dB.

CH<x>:PROBEFunc:EXTUnits

Description

This command sets the unit of measurement for the external attenuator of the specified channel. The channel is specified by x, which ranges 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 measure 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

Related Commands

CH<x>:PRObe:UNIts? (see page 445)

Syntax 1

```
CH<x>:PROBEFunc:EXTUnits <str>
```

Syntax 2

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

Argument

- `<str>`

  This can contain a string of up to eight characters to indicate the attenuation unit of measurement for the specified channel. However, most oscilloscope attenuators only display the first two characters.

Example 1

```
CH4:PROBEFunc:EXTUnits "Pa"
```

This command sets the unit of measurement for the Channel 4 external attenuator.

Example 2

```
CH2:PROBEFunc:EXTUnits?
```

This query might return `:CH2:PROBEFUNC:EXTUNITS "Pa"`, indicating that the Channel 2 external attenuator units of measurement are pascals.

CH<x>:SCAle

Description

This command sets or queries the vertical scale of the specified channel. The channel is specified by x, which 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

Related Commands

CH<x>OFFSet (see page 436), CH<x>:POSition (see page 437), REF<x>:SCAle (see page 458), MATH<x>:SCAle (see page 207)

Syntax 1

```
CH<x>:SCAle <NR3>
```

Syntax 2

```
CH<x>:SCAle?
```

Argument

- `<NR3>`

  This is the vertical channel scale in units per division.

Example 1

```
CH4:SCAle 100E-03
```

This command sets the channel 4 scale to 100 mV per division.

Example 2

```
CH2:SCAle?
```

This query might return `:CH2:SCALE 1.0000E+00`, indicating that the current scale setting of channel 2 is 1 volt per division.

CH<x>:TERmination

Description

This command sets the connected/disconnected status of a 50 ohm resistor, which may be connected between the specified channel's coupled input and oscilloscope ground. The channel is specified by x, which ranges 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.

Group

Vertical

Syntax 1

```
CH<x>:TERmination <NR3>
```

Syntax 2

```
CH<x>:TERmination?
```

Argument

- `<NR3>`

  This specifies the channel input resistance, which can be specified as 50 or 1,000,000.

Example 1

```
CH4:TERmination 50.0E+0
```

This command establishes 50 ohm impedance on channel 1.

Example 2

```
CH2:TERmination?
```

This query might return `:CH2:TERMINATION 1.0000E+06`, indicating that channel 2 is set to 1 MegOhm.

MATH<x>:POSition?

Description

This command sets or queries the vertical position of the specified Math waveform. The Math waveform is specified by x, which can range 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 channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal 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.

Groups

Math, Vertical

Related Commands

CH<x>POSition (see page 437), REF<x>:POSition (see page 457)

Syntax 1

```
MATH<x>:POSition <NR3>
```

Syntax 2

```
MATH<x>:POSition?
```

Argument

- `<NR3>`

  This is the desired position in divisions from the center graticule.

Example 1

```
MATH2:POSition 1.3E+00
```

This command positions the Math 2 input signal 1.3 divisions higher than a position of 0.

Example 2

```
MATH1:POSition?
```

This query might return `:MATH1:POSITION -1.3000E+00`, indicating that the current position of Math 1 is 1.3 divisions below the center graticule.

MATH<x>:SCAle

Description

This command sets or queries the vertical scale of the specified math waveform. The math waveform is specified by x, which can range 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 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. For reference and math waveforms, the scale setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

In addition to using the MATH<x>:SCAle command, autoscaling occurs when a math is first defined and enabled, or when a math string changes. After the math is calculated for the first time, the instrument determines the min + max of that waveform data. Then, the math position is set 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.

Group

Math, Vertical

Related Commands

CH<x>:SCAle (see page 449), REF<x>:SCAle (see page 458)

Syntax 1

```
MATH<x>:SCAle <NR3>
```

Syntax 2

```
MATH<x>:SCAle?
```

Argument

- `<NR3>`

  This is the scale, in volts, amps or watts per division. The range is from 100.0E-36 through 100.0E+36.

Example 1

```
MATH4:SCAle 100E-03
```

This command sets the Math 4 scale to 100 mV per division.

Example 2

```
CH2:SCAle?
```

This query might return `:MATH2:SCALE 1.0000E+00`, indicating that the current scale setting of Math 2 is 1 volt per division.

REF<x>:LABel?

Description

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. References are specified by x, which ranges 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

Related Commands

REF<x>:LABel:NAMe (see page 454), REF<x>:LABel:XPOS (see page 455), REF<x>:LABel:YPOS (see page 456)

Syntax

```
REF<x>:LABel?
```

Argument

None.

Example

```
REF1:LABel?
```

This query will return `:REF1:LABel:NAMe "Myname";:XPOS-200;:YPOS 50`.

REF<x>:LABel:NAMe

Description

This command sets or queries the label of the designated waveform. References are specified by x, which ranges from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Reference Setup from the drop-down list. Then select the tab associated with the reference for which you want to create a label and then enter a label in the Label box.

Group

Vertical

Related Commands

REF<x>:LABel? (see page 453)

Syntax 1

```
REF<x>:LABel:NAMe <String>
```

Syntax 2

```
REF<x>:LABel:NAMe?
```

Argument

- `<String>`

  This is the character string that will be used for the reference waveform label name.

Example 1

```
REF3:LABel:NAMe?
```

This query might return `:REF3:LABEL:NAME "Signal2"`, indicating that the label name for Reference 3 is currently set to "Signal2".

Example 2

```
REF4:LABel:NAMe "My Reference"
```

This command sets the label name of Reference 4 to "My Reference".

REF<x>:LABel:XPOS

Description

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. References are specified by x, which ranges from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Reference Setup from the drop-down list. Then either view or set X Position.

Group

Vertical

Related Commands

REF<x>:LABel? (see page 453)

Syntax 1

```
REF<x>:LABel:XPOS <NR1>
```

Syntax 2

```
REF<x>:LABel:XPOS?
```

Argument

- `<NR1>`

  This 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.

Example 1

```
REF4:LABel:XPOS 10
```

This command moves the waveform label for the REF3 waveform, so that it begins 10 pixels to the right of the left edge of the screen.

Example 2

```
REF2:LABel:XPOS?
```

This query 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

Description

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. References are specified by x, which ranges from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Reference Setup from the drop-down list. Then either view or set Y Position.

Group

Vertical

Related Commands

REF<x>:LABel? (see page 453)

Syntax 1

```
REF<x>:LABel:YPOS <NR1>
```

Syntax 2

```
REF<x>:LABel:YPOS?
```

Argument

This 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 400.

Example 1

```
REF3:LABel:YPOS -10
```

This command moves the waveform label for the REF3 waveform to just beneath (10 pixels) the top of the screen.

Example 2

```
REF2:LABel:YPOS?
```

This query might return `:REF2:LABEL:YPOS 0`, indicating that the waveform label for the REF2 waveform is currently located just beneath the top of the screen.

REF<x>:POSition

Description

This command sets or queries the vertical position of the specified reference waveform. The reference waveform is specified by x, which ranges from 1 through 4. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup from the drop-down list and then entering the desired Position value.

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

Related Commands

CH<x>:POSition (see page 437), MATH<x>:POSition (see page 206)

Syntax 1

```
REF<x>:POSition <NR3>
```

Syntax 2

```
REF<x>:POSition?
```

Argument

- `<NR3>`

  This is the desired position, in divisions from the central graticule. The range is from -5 to 5 divisions.

Example 1

```
REF2:POSition 1.3E+00
```

This command positions the Reference 2 input signal 1.3 divisions above the center graticule.

Example 2

```
REF1:POSition?
```

This query might return `:REF1:POSITION -1.3000E+00`, indicating that the current position of Reference 1 is 1.3 divisions below the center graticule.

REF<x>:SCAle

Description

This command sets or queries the vertical scale of the specified reference waveform. The reference is specified by x, which can range from 1 through 4.

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

Related Commands

CH<x>:SCAle (see page 449), MATH<x>:SCAle (see page 207)

Syntax 1

```
REF<x>:SCAle <NR3>
```

Syntax 2

```
REF<x>:SCAle?
```

Argument

- `<NR3>`

  This is the gain, in user units per division..

Example 1

```
REF4:SCAle 100E-03
```

This command sets the Reference 4 scale to 100 mV per division.

Example 2

```
CH2:SCAle?
```

This query might return `:REF2:SCALE 1.0000E+00`, indicating that the current scale setting of Reference 2 is 1 volt per division.

SELect?

Description

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?
```

Example

```
SELect?
```

This query might return `: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:<wfm>

Description

This command turns on a specified waveform. This action also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

Group

Vertical

Syntax 1

```
SELect:<wfm> {ON|OFF|<NR1>}
```

Syntax 2

```
SELect:<wfm>?
```

Arguments

- `ON`

  This turns on the display of the specified waveform. This waveform also becomes the selected waveform.

- `OFF`

  This turns off the display of the specified waveform.

- `<NR1>`

  A 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

Example

```
SELect:CH2 ON
```

This command turns the channel 2 display on and selects channel 2.

```
SELect:REF1?
```

This query might return `:SELECT:REF1`, indicating that REF1 is the selected waveform.

SELect:CONTRol <wfm>

Description

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 1

```
SELect:CONTROl (CH<x>|MATH<x>|REF<x>)
```

Syntax 2

```
SELect:CONTROl?
```

Arguments

- `CH<x>`

  This selects the specified channel waveform as the waveform that is affected by the front-panel controls. The valid channel waveform range is from 1 through 4.

- `MATH<x>`

  This selects the specified math waveform as the waveform that is affected by the front-panel controls. The valid math waveform range is from 1 through 4.

- `REF<x>`

  This selects the specified reference waveform as the waveform that is affected by the front-panel controls. The valid reference waveform range is from 1 through 4.

Example 1

```
SELect:CONTROl CH2
```

This command resets acquisition, displays channel 2, and causes the selected waveform to be the implied object of waveform-manipulating commands.

Example 2

```
SELect:CONTROl?
```

This query might return `:SELECT:CONTROL MATH3,` indicating that math 3 is the implied object of waveform-manipulating commands.

Waveform Transfer Command Group

Waveform Transfer Overview

You 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 oscilloscope 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 with a MATH) is involved, 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 IBM-compatible 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 IBM-compatible 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 WFMInopre: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 oscilloscope. You can only transfer one waveform at a time. Waveforms sent to the instrument are always stored in one of the four reference memory locations. You 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 Oscilloscope

You can transfer waveforms from the oscilloscope 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 oscilloscope using CURVe?.

Transferring Waveform Data to the Oscilloscope

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.

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

Waveform Transfer Commands

Command	Description
CURve?	Transfers waveform data from instrument specified by the DATa:SOUrce command
CURVe	Transfer waveform data to the instrument (reference memory location specified by DATa:DESTination)
DATa?	Returns waveform data format and location
DATa	Sets the waveform data format and location
DATa:DESTination?	Returns the reference memory location for storing waveform data sent to the oscilloscope
DATa:DESTination	Sets the reference memory location for storing waveform data sent to the oscilloscope
DATa:ENCdg?	Returns the format of outgoing waveform data
DATa:ENCdg	Sets the format of outgoing waveform data
DATa:SOUrce?	Returns the location of waveform data transferred from the instrument
DATa:SOUrce	Sets the location of waveform data transferred from the instrument
DATa:STARt?	Returns the starting point in waveform transfer
DATa:STARt	Sets the starting point in waveform transfer
DATa:STOP?	Returns the ending data point in waveform transfer
DATa:STOP	Sets the ending data point in waveform transfer
WAVFrm?	Returns a branch query containing waveform data in either binary or ASCII format, waveform formatting data, and the location of the waveform data source
WFMInpre?	Returns the waveform formatting specification to be applied to the next incoming CURVE command data
WFMInpre:BIT_Nr?	Returns the number of bits per binary waveform point
WFMInpre:BIT_Nr	Sets the number of bits per binary waveform point
WFMInpre:BN_Fmt?	Returns the format of binary data for the incoming waveform
WFMInpre:BN_Fmt	Sets the format of binary data for the incoming waveform
WFMInpre:BYT_Nr?	Returns the binary field data width for the first ordered waveform
WFMInpre:BYT_Nr	Sets the binary field data width for the first ordered waveform
WFMInpre:BYT_Or?	Returns the byte order of waveform points
WFMInpre:BYT_Or	Sets the byte order of waveform points

WFMInpre:ENCdg?	Returns the type of encoding for incoming waveform data
WFMInpre:ENCdg	Sets the type of encoding for incoming waveform data
WFMInpre:NR_Pt?	Returns the number of points in the transmitted waveform record
WFMInpre:NR_Pt	Sets the number of points in the transmitted waveform record
WFMInpre:PT_Fmt?	Returns the point format of incoming waveform data
WFMInpre:PT_Fmt	Sets the point format of incoming waveform data
WFMinpre:PT_OFF?	Returns the trigger point within the incoming waveform data
WFMinpre:PT_OFF	Sets the trigger point within the incoming waveform data
WFMinpre:WFID	Provided to allow a waveform extracted from the oscilloscope to be more easily sent back
WFMInpre:XINcr?	Returns the horizontal sampling interval between incoming waveform points
WFMInpre:XINcr	Sets the horizontal sampling interval between incoming waveform points
WFMInpre:XUNit?	Returns the horizontal units of the incoming waveform
WFMInpre:XUNit	Sets the horizontal units of the incoming waveform
WFMInpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMInpre:XZEro	Sets the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMInpre:YMUlt?	Returns the vertical scale factor, per digitizing level, of the incoming waveform points
WFMInpre:YMUlt	Sets the vertical scale factor, per digitizing level, of the incoming waveform points
WFMInpre:YOFf?	Returns the vertical position of the incoming waveform in digitizing levels
WFMInpre:YOFf	Sets the vertical position of the incoming waveform in digitizing levels
WFMInpre:YUNit?	Returns the vertical units of the incoming waveform
WFMInpre:YUNit	Sets the vertical units of the incoming waveform
WFMInpre:YZEro?	Returns the offset of the incoming waveform
WFMInpre:YZEro	Sets the offset of the incoming waveform
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command
WFMOutpre:BIT_Nr?	Returns the number of bits per waveform

	point that outgoing waveforms contain
WFMOutpre:BIT_Nr	Sets the number of bits per waveform point that outgoing waveforms contain
WFMOutpre:BN_Fmt?	Returns the format of binary data for the waveform
WFMOutpre:BN_Fmt	Sets the format of binary data for the waveform
WFMOutpre:BYT_Nr?	Returns the data width for the waveform
WFMOutpre:BYT_Nr	Sets the data width for the waveform
WFMOutpre:BYT_Or?	Returns the byte order of waveform points
WFMOutpre:BYT_Or	Sets the byte order of waveform points
WFMOutpre:ENCdg?	Returns the type of encoding for outgoing waveforms
WFMOutpre:ENCdg	Sets 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_OR?	Returns whether the source waveform is DPO
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

CURVe

Description

This command transfers waveform data to and from the oscilloscope. 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 that 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

Related Commands

DATa:DESTination (see page 469), DATa:SOUrce (see page 472), DATa:STARt (see page 473), DATa:STOP (see page 474), WFMInpre? (see page 476), WFMInpre:BYT_Nr (see page 480), WFMOutpre? (see page 493)

Syntax 1

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

Syntax 2

```
CURVe?
```

Arguments

- `<Block>`

  This 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>`

  This is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>…], where each <NR1> represents a data point.

Example 1

```
CURVe?
```

This query 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`

Example 2

```
CURVe <Block>
```

This command sets the format of the waveform data, transferred to and from the oscilloscope, to binary format.

DATa

Description

This command sets or queries the format and location of the waveform data that is transferred with the CURVe command.

Group

Waveform Transfer

Related Commands

CURVe (see page 467), DATa:STARt (see page 473), DATa:STOP (see page 474), DATa:ENCdg (see page 470)

Syntax 1

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

Syntax 2

```
DATa?
```

Argument

- `INIT`

  This initializes the waveform data parameters to their factory defaults.

- `SNAp`

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

Example 1

```
DATa?
```

This query might return `:DATA:DESTINATION REF1:ENCDG RIBINARY;SOURCE CH1;START 1;STOP 500;WIDTH 1`

Example 2

```
DATa INIT
```

This command initializes the waveform data parameters to their factory defaults.

DATa:DESTination

Description

This command sets or queries the reference memory location for storing waveform data that is transferred into the oscilloscope by the CURVe command.

Group

Waveform Transfer

Related Commands

CURVe (see page 467)

Syntax 1

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

Syntax 2

```
DATa:DESTination?
```

Argument

- `REF<x>`

  This is the reference where the waveform will be stored. The reference number is specified by x, which ranges from 1 through 4.

Example 1

```
DATa:DESTination?
```

This query might return `:DATA:DESTINATION REF3`, indicating that reference 3 is the currently selected reference memory location for waveform data.

Example 2

```
DATa:DESTination REF1
```

This command requests that incoming waveform data be stored in reference 1.

DATa:ENCdg

Description

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

Related Commands

WFMOutpre:ENCdg (see page 494), WFMOutpre:BN_Fmt (see page 496), WFMOutpre:BYT_Or (see page 497)

Syntax 1

```
DATa:ENCdg {ASCIi|FAStest|RIBinary|RPBinary|FPBinary|SRIbinary
|SRPbinary|SFPbinary}
```

Syntax 2

```
DATa:ENCdg?
```

Argument

- `ASCIi`

  This specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN_Fmt and :BYT_Or are ignored.

- `FAStest`

  This requests 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 MSB, 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 FP format will be used.

- `RIBinary`

  This 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. 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`

  This specifies the positive integer datapoint 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. Center 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`

  This specifies the floating point (width = 4) data.

  The range is from -3.4×10^{38} to 3.4×10^{38}. Center 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.

- `SRIbinary`

  This 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`

  This 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 IBM compatible PCs.

- `SFPbinary`

  This specifies floating point data in IBM format.

DATa and WFMOUTPRE Parameter Settings				
	WFMOUTPRE Settings			
DATa:ENCdg Setting	:ENCdg	:BN_Fmt	:BYT_Or	:BYT_NR
ASCii	ASC	N/A	N/A	1, 2, 4, 8
FAStext	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

Example 1

```
DATa:ENCdg?
```

This query might return `:DATa:ENCDG SRPBINARY` for the format of the outgoing waveform data.

Example 2

```
DATa:ENGdg RPBinary
```

This command sets the data encoding format to be a positive integer where the most significant byte is transferred first.

DATa:SOUrce

Description

This command sets or queries the location of waveform data that is transferred from the oscilloscope by the CURVe? Query.

Group

Waveform Transfer

Related Commands

CURVe? (see page 467)

Syntax 1

```
DATa:SOUrce <wfm>[<,><wfm>]
```

Syntax 2

```
DATa:SOUrce?
```

Argument

- `<wfm>`

  This is the location of the waveform data that will be transferred from the oscilloscope to the controller.

Example 1

```
DATa:SOUrce?
```

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

Example 2

```
DATa:SOUrce CH1
```

This command specifies that the CH1 waveforms will be transferred in the next CURVe? query.

DATa:STARt

Description

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 oscilloscope.

Group

Waveform Transfer

Related Commands

DATa (see page 468), DATa:STOP (see page 474), WFMInpre:NR_Pt (see page 482), WFMOutpre:NR_Pt (see page 499)

Syntax 1

```
DATa:STARt <NR1>
```

Syntax 2

```
DATa:STARt?
```

Argument

- `<NR1>`

  This 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.

  When DATa:STOP is less than DATa:STARt, the stop value will equal DATa:STARt + (DATa:STARt - DATa:STOP). For example, if DATa:STARt = 30 and DATa:STOP = 20, then the range of data points for the waveform transfer will equal 30 through 40.

Example 1

```
DATa:STARt?
```

This query might return `:DATA:START 214`, indicating that this is the first waveform data point that will be transferred.

Example 2

```
DATa:STARt 10
```

This command specifies that the waveform transfer will begin with data point 10.

DATa:STOP

Description

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.

Note: When using the CURVe command, DATa:STOP is ignored and WFMInpre:NR_Pt is used.

Group

Waveform Transfer

Related Commands

CURVe (see page 467), DATa (see page 468), WFMInpre:NR_Pt (see page 482)

Syntax 1

```
DATa:STOP <NR1>
```

Syntax 2

```
DATa:STOP?
```

Argument

- `<NR1>`

  This 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.

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

  When DATa:STOP is less than DATa:STARt, the stop value will equal DATa:STARt + (DATa:STARt - DATa:STOP). For example, if DATa:STARt = 30 and DATa:STOP = 20, then the range of data points for the waveform transfer will equal 30 through 40.

Example 1

```
DATa:STOP?
```

This query might return `:DATA:STOP 14900`, indicating that this is the last waveform data point that will be transferred.

Example 2

```
DATa:STOP 15000
```

This command specifies that the waveform transfer will stop at data point 15000.

WAVFrm?

Description

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

Related Commands

CURVe (see page 467), DATa:SOUrce (see page 472), WFMOutpre? (see page 493)

Syntax

```
WAVFrm?
```

Example

```
WAVFrm?
```

This query might return:

```
: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,
Sample mode";:CURVE 51,50,51,48,51,48,50,49,51,49,51,48,
51,48,51,49,50,49,50,48,49,49,52,49,49,50,50,48,50,49,
49,49,49,49,50,47,49,47,50,48,49,48,50,48,49,47,49,48,
51,48,49,47,50,48,50,47,51,47,49,48,48,48,50,46,50,46,
48,45,48,47,49,47,49,48,49,48,49,45,49,47,48,46,48,48,
49,45,49,45,47,46,46,46,49,47,48,46,48,46,48,47,47,46,
47,47,48,46,48,45,48,46,47,46,47,47,46,45,46,45,47,47,
47,46,46,44,47,45,45,45,46,43,46,45,46,44,46,45,48,45,
45,44,46,45,46,45,45,45,45,43,45,44,46,44,47,44,44,44,
45,43,44,44,45,44,46,43,46,43,45,44,45,44,45,41,44,43,
45,42,44,43,44,43,44,42,43,42,44,41,44,41,44,42,43,42,
43,42,45,42,43,41,42,41,43,42,44,42,43,42,43,40,42,40,
42,40,42,41,43,41,42,41,42,39,41,41,42,42,43,41,42,39,
41,39,41,40,41,40,41,39,41,38,41,39,41,38,42,39,41,37,
40,38,40,38,39,38,39,38,41,39,39,38,40,37,38,37,38,38,
39,38,40,36,39,37,39,37,39,37,39,36,40,38,38,35,38,36,
38,37,39,37,38,37,38,37,37,36,38,35,38,36,37,36,38,36,
38,35,39,34,36,36,36,35,38,35,36,34,37,34,37,35,37,36,
37,35,37,35,36,34,36,33,36,35,36,34,36,33,34,33,34,34,
34,34,36,33,36,34,36,33,34,33,34,32,34,32,34,33,33,31,
33,32,34,32,33,32,32,31,33,31,31,32,33,31,32,31,32,30,
32,30,32,31,31,30,31,30,31,29,31,29,32,31,31,30,30,29,
30,29,30,30,31,29,30,28,30,29,31,28,30,28,30,29,29,27,
29,28,29,27,30,27,30,26,28,28,29,27,28,27,28,27,28,28,
28,27,28,27,28,26,28,26,28,26,26,25,27,25,28,27,28,26,
25,25,26,25,25,25,25,25,26,25,27,25,25,24,25,25,26,24,
26,24,24,23,25,24,24,22,25,23,24,23,24,22,24,22,23,22,
25,24,24,21,23,22,22,21,24,22,22,22,22,21,22,20,23,21,
21,20
```

WFMInpre?

Description

This query-only command returns the waveform formatting specification to be applied to the next incoming CURVe command data.

Group

Waveform Transfer

Related Commands

CURVe (see page 467), WFMOutpre? (see page 493)

Syntax

```
WFMInpre?
```

Example

```
WFMInpre?
```

This query might return `:WFMINPRE:BIT_NR 8;BN_FMT RI;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:ENCdg

Description

This command sets or queries the type of encoding for incoming waveform data.

Group

Waveform Transfer

Related Commands

WFMOutpre:ENCdg (see page 494)

Syntax 1

```
WFMInpre:ENCdg {ASC|BIN}
```

Syntax 2

```
WFMInpre:ENCdg?
```

Arguments

- `ASC`

  This specifies that the incoming data is in ASCII format.

- `BIN`

  This 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.

Example 1

```
WFMInpre:ENCdg?
```

This query might return `:WFMINPRE:ENCDG BIN`, indicating that the incoming waveform data will be in binary format.

Example 2

```
WFMInpre:ENCdg ASC
```

This command sets the format of incoming waveform data to ASCII format.

WFMInpre:BN_Fmt

Description

This command sets or queries the format of binary data for incoming waveforms.

Group

Waveform Transfer

Related Commands

WFMOutpre:BN_Fmt (see page 496)

Syntax 1

```
WFMInpre:BN_Fmt {RI|RP|FP}
```

Syntax 2

```
WFMInput:BN_Fmt?
```

Arguments

- `RI`

  This specifies signed integer datapoint representation.

- `RP`

  This specifies positive integer datapoint representation.

- `FP`

  This specifies single-precision binary floating point representation.

Example 1

```
WFMInpre:BN_Fmt?
```

This query might return `:WFMINPRE:BN_FMT RI`, indicating that the incoming data is currently interpreted as signed integers.

Example 2

```
WFMInpre:BN_Fmt FP
```

This command specifies that incoming data will be interpreted as single-precision binary floating point.

WFMInpre:BYT_Or

Description

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

Related Commands

WFMInpre:ENCdg (see page), WFMInpre:BYT_Nr (see page 480), WFMOutpre:BYT_Or (see page 497)

Syntax 1

```
WFMInpre:BYT_Or {LSB|MSB}
```

Syntax 2

```
WFMInpre:BYT_Or?
```

Arguments

- `LSB`

  This specifies that the least significant byte will be transmitted first.

- `MSB`

  This specifies that the most significant byte will be transmitted first.

Example 1

```
WFMInpre:BYT_Or?
```

This query might return `:WFMINPRE:BYT_OR LSB`, indicating that the least significant incoming CURVe data byte will be transmitted first.

Example 2

```
WFMInpre:BYT_Or MSB
```

This command sets the most significant incoming byte of incoming waveform data to be transmitted first.

WFMInpre:BYT_Nr

Description

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

Related Commands

DATa:DESTination (see page 469), WFMInpre:BN_Fmt (see page 478), WFMInpre:ENCdg (see page 477), WFMInpre:BIT_Nr (see page 481), WFMOutpre:BYT_Nr (see page 498)

Syntax 1

```
WFMInpre:BYT_Nr <NR1>
```

Syntax 2

```
WFMInpre:BYT_Nr?
```

Argument

- `<NR1>`

  This is the number of bytes per data point and can be 1, 2 (RI, RP) or 4 (FP).

Example 1

```
WFMInpre:BYT_Nr?
```

This query might return `:WFMINPRE:BYT_NR 2`, indicating that there are 2 bytes per incoming waveform data point.

Example 2

```
WFMInpre:BYT_Nr 1
```

This command sets the number of bytes per incoming waveform data point to 1, which is the default setting.

WFMInpre:BIT_Nr

Description

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

Related Commands

DATa:DESTination (see page 469), WFMInpre:ENCdg (see page 477), WFMInpre:BYT_Nr (see page 480), WFMOutpre:BIT_Nr (see page 495)

Syntax 1

```
WFMInpre:BIT_Nr <NR1>
```

Syntax 2

```
WFMInpre:BIT_Nr?
```

Argument

- `<NR1>`

  This number of bytes per data point can be 8, 16 (RI, RP) or 32 (FP).

Example 1

```
WFMInpre:BIT_Nr?
```

This query might return `:WFMINPRE:BIT_NR 8`, indicating that incoming RI or RP binary format data uses 8 bits per waveform point.

Example 2

```
WFMInpre:BIT_Nr 16
```

This command sets the number of bits per waveform point to 16, for incoming RI and RP binary format data.

WFMInpre:NR_Pt

Description

This command sets or returns the number of points that are in the transmitted waveform record.

Group

Waveform Transfer

Related Commands

WFMInpre:PT_Fmt (see page 483), WFMOutpre:NR_Pt? (see page 499)

Syntax 1

```
WFMInpre:NR_Pt <NR1>
```

Syntax 2

```
WFMInpre:NR_Pt?
```

Argument

- `<NR1>`

  If WFMInpre:PT_Fmt is set to Y, this is the number of data points; If WFMInpre:PT_Fmt is set to ENV, this is the number of max-min pairs.

Example 1

```
WFMInpre:NR_Pt?
```

This query might return `:WFMINPRE:NR_PT 8000`, indicating that there are 8000 data points in the transmitted waveform record.

Example 2

```
WFMInpre:NR_Pt 5000
```

This command specifies that 5000 data points will be expected.

WFMInpre:PT_Fmt

Description

This command sets or queries the point format of the incoming waveform data. Regardless of the argument used, the scale, offset, etc. are interpreted similarly. When ENV is used, waveform data interpreted over the min/max pair; when Y is used, it is interpreted over a single point.

Group

Waveform Transfer

Related Commands

WFMOutpre:PT_Fmt? (see page 500)

Syntax 1

```
WFMInpre:PT_Fmt {ENV|Y}
```

Syntax 2

```
WFMInpre:PT_Fmt?
```

Arguments

- `ENV`

  This specifies that the waveform is transmitted 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`

  This 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)

Example 1

```
WFMInpre:PT_Fmt ENV
```

This command sets the incoming waveform data point format to enveloped.

Example 2

```
WFMInpre:PT_Fmt?
```

This query might return `:WFMINPRE:PT_FMT ENV`, indicating that the waveform is transmitted as maximum and minimum point pairs.

WFMInpre:PT_Off

Description

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

Related Commands

DATa:DESTination (see page 469), DATa:STARt (see page 473), WFMOutpre:PT_Off (see page 501)

Syntax 1

```
WFMInpre:PT_Off <NR1>
```

Syntax 2

```
WFMInpre:PT_Off?
```

Argument

- `<NR1>`

  This 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.

Example 1

```
WFMInpre:PT_Off 0
```

This command specifies that the trigger point is the first point in the waveform record, which is the default.

Example 2

```
WFMInpre:PT_Off?
```

This query might return `:WFMINPRE:PT_OFF 0`, indicating the incoming waveform trigger point.

WFMInpre:WFId

Description

This command (no query form) accepts but ignores the argument. This command is provided only to allow a waveform extracted from the oscilloscope to be more easily imported.

Group

Waveform Transfer

Related Commands

DATa:DESTination (see page 469), WFMInpre:BN_Fmt (see page 478), WFMInpre:ENCdg (see page 477), WFMOutpre:WFid (see page 503)

Syntax

```
WFMInpre:WFId <String>
```

Argument

- `<String>`

  This must be a valid 488.2 string (but the contents are ignored).

Example

```
WFMInpre:WFId "Ch1, DC coupling, 2.000V/div, 400.0ns/div, 500 points, Sample mode"
```

This is a syntactically correct command.

WFMInpre:XINcr

Description

This command sets or queries the horizontal interval between incoming waveform points in units specified by WFMInpre:XUNit.

Group

Waveform Transfer

Related Commands

WFMInpre:XUNit (see page 488), WFMOutpre:XINcr (see page 505)

Syntax 1

```
WFMInpre:XINcr <NR3>
```

Syntax 2

```
WFMInpre:XINcr?
```

Argument

- `<NR3>`

  This is the horizontal interval representation.

Example 1

```
WFMInpre:XINcr?
```

This query might return `:WFMINPRE:XINCR 1.0000E-3`, indicating that (assuming that WFMInpre:XUNit is set to "s") there is a 1 ms interval between incoming waveform points.

Example 2

```
WFMInpre:XINcr 3E-3
```

This command sets the interval between Incoming waveform points to 3 ms.

WFMInpre:XZEro

Description

This command sets or queries the (subsample) time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger of 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

Related Commands

WFMInpre:PT_Off (see page 484), WFMInpre:XINcr (see page 486), WFMInpre:XUnit (see page 488), WFMOutpre:XZEro (see page 506)

Syntax 1

```
WFMInpre:XZEro <NR3>
```

Syntax 2

```
WFMInpre:XZEro?
```

Argument

- `<NR3>`

  This is value that ranges from -WFMInpre:XINcr to 0.

Example 1

```
WFMInpre:XZEro?
```

This query might return `:WFMINPRE:XZEro 7.5000E-6`, indicating that the trigger occurs 7.5 µs before the sample designated by WFMInpre:PT_Off.

Example 2

```
WFMInpre:XZEro 5.7E-6
```

This command specifies that the trigger actually occurred 5.7 µs before the sample designated by WFMInpre:PT_Off .

WFMInpre:XUNit

Description

This command sets or returns the horizontal units of the incoming waveform.

Group

Waveform Transfer

Related Commands

WFMOutpre:XUNit (see page 507)

Syntax 1

```
WFMInpre:XUNit <String>
```

Syntax 2

```
WFMInpre:XUNit?
```

Argument

- `<String>`

  This contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

Example 1

```
WFMInpre:XUNit?
```

This query might return `:WFMINPRE:XUNIT "s"`, indicating that the horizontal units for the incoming waveform are expressed in seconds.

Example 2

```
WFMInpre:XUNit "Hz"
```

This command specifies that the horizontal units for the incoming waveform are hertz.

WFMInpre:YMUlt

Description

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

Related Commands

DATa:DESTination (see page 469), WFMInpre:BYT_Nr (see page 480), WFMInpre:YUNit (see page 488)

Syntax 1

```
WFMInpre:YMUlt <NR3>
```

Syntax 2

```
WFMInpre:UMUlt?
```

Argument

- `<NR3>`

  This is the vertical scale factor per digitizing level of the incoming waveform points.

Example 1

```
WFMInpre:YMUlt?
```

This query might return `:WFMINPRE:YMULT 40.0000E-3`, indicating that the vertical scale is 40 mV/digitizing level (1V/div).

Example 2

```
WFMInpre:YMUlt 20E-3
```

This command 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:YOFf

Description

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 nonfloating point formats), this command must take the location of the binary point implied by BYT_NR into consideration.

Group

Waveform Transfer

Related Commands

WFMInpre:BYT_Nr (see page 480), WFMInpre:YMUlt? (see page 489), WFMOutpre:YOFf? (see page 509)

Syntax 1

```
WFMInpre:YOFf <NR3>
```

Syntax 2

```
WFMInpre:YOFf?
```

Argument

- `<NR3>`

  This is the vertical offset in digitizing levels.

Example 1

```
WFMInpre:YOFf?
```

This query might return `:WFMINPRE:YOFF 25`, indicating the vertical position of the incoming waveform in digitizing levels.

Example 2

```
WFMInpre:YOFf 50
```

This command specifies that the zero reference for the incoming waveform is 50 digitizing levels (2 divisions) above the center of the data range.

WFMInpre:YUNit

Description

This command sets or queries the vertical units of the incoming waveform.

Group

Waveform Transfer

Related Commands

WFMOutpre:YUNit (see page 510)

Syntax 1

```
WFMInpre:YUNit <String>
```

Syntax 2

```
WFMInpre:YUNit?
```

Argument

- `<String>`

  This contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.

Example 1

```
WFMInpre:YUNit?
```

This query might return `:WFMINPRE:YUNIT "s"`, indicating the vertical units for the incoming waveform are expressed in seconds.

Example 2

```
WFMInpre:YUNit "Pa"
```

This command specifies that the vertical units for the incoming waveform are pascal.

WFMInpre:YZEro

Description

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

Related Commands

WFMInpre:YUNit (see page 491), WFMOutpre:YZEro (see page 511)

Syntax 1

```
WFMInpre:YZEro <NR3>
```

Syntax 2

```
WFMInpre:YZEro?
```

Argument

- `<NR3>`

  This is of the offset in YUNits,

Example 1

```
WFMInpre:YZEro?
```

This query 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).

Example 2

```
WFMInpre:YZEro 1.5E+0
```

This command specifies that the zero reference for the incoming waveform is 1.5 V below the center of the data range (given that WFMInpre:YUNit is set to V).

WFMOutpre?

Description

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?
```

Example

```
WFMOutpre:?
```

This query might return

```
: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:ENCdg

Description

This command sets and queries the type of encoding for outgoing waveforms.

Group

Waveform Transfer

Related Commands

DATa:ENCdg (see page 470), WFMOutpre:BYT_Nr (see page 498), WFMOutpre:BYT_Or (see page 497), WFMOutpre:BIT_Nr (see page 495), WFMOutpre:BN_Fmt (see page 496)

Syntax 1

```
WFMOutpre:ENCdg {ASC|BIN}
```

Syntax 2

```
WFMOutpre:ENCdg?
```

Argument

- `ASC`

  This 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 floats will be sent as <NR3> numbers.

- `BIN`

  This 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.

Example 1

```
WFMOutpre:ENCdg?
```

This query might return `:WFMOUTPRE:ENCDG BIN`, indicating that outgoing waveform data will be sent in binary format.

Example 2

```
WFMOutpre:ENCdg ASC
```

This command specifies that the outgoing waveform data will be sent in ASCII format.

WFMOutpre:BIT_Nr

Description

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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:BN_Fmt (see page 478), WFMOutpre:ENCdg (see page 494)

Syntax 1

```
WFMOutpre:BIT_Nr <NR1>
```

Syntax 2

```
WFMOutpre:BIT_Nr?
```

Argument

- `<NR1>`

  This number of bits per data point can be 8, 16, 32 or 64.

Example 1

```
WFMOutpre:BIT_Nr?
```

This query might return `:WFMOUTPRE:BIT_NR 8`, indicating that outgoing RI or RP binary; format data uses 8 bits per waveform point.

Example 2

```
WFMOutpre:BIT_Nr 16
```

This command sets the number of bits per waveform point to 16, for incoming RI and RP binary format data.

WFMOutpre:BN_Fmt

Description

This command sets or queries the format of binary data for outgoing waveforms specified by the DATa:SOUrce command.

Group

Waveform Transfer

Related Commands

DATa:SOUrce (see page 472)

Syntax 1

```
WFMOutpre:BN_Fmt {RI|RP|FP}
```

Syntax 2

```
WFMOutput:BN_Fmt?
```

Arguments

- `RI`

  This specifies signed integer datapoint representation.

- `RP`

  This specifies positive integer datapoint representation.

- `FP`

  This specifies single-precision binary floating point.

Example 1

```
WFMOutpre:BN_Fmt?
```

This query might return `:WFMOUTPRE:BN_FMT RI`, indicating that the outgoing waveform data is currently in signed integer format.

Example 2

```
WFMOutpre:BN_Fmt FP
```

This command specifies that outgoing waveform data will be in single-precision binary floating point format.

WFMOutpre:BYT_Or

Description

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

Related Commands

WFMOutpre:ENCdg (see page 494)

Syntax 1

```
WFMOutpre:BYT_Or {LSB|MSB}
```

Syntax 2

```
WFMOutpre:BYT_Or?
```

Arguments

- `LSB`

  This specifies that the least significant byte will be transmitted first.

- `MSB`

  This specifies that the most significant byte will be transmitted first.

Example 1

```
WFMOutpre:BYT_Or?
```

This query might return `:WFMOUTPRE:BYT_OR LSB`, indicating that the least significant data byte will be transmitted first.

Example 2

```
WFMOutpre:BYT_Or MSB
```

This command sets the most significant outgoing byte of waveform data to be transmitted first.

WFMOutpre:BYT_Nr

Description

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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:BN_Fmt (see page 496), WFMOutpre:ENCdg (see page 494)

Syntax 1

```
WFMOutpre:BYT_Nr <NR1>
```

Syntax 2

```
WFMOutpre:BYT_Nr?
```

Argument

- `<NR1>`

  This 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.

Example 1

```
WFMOutpre:BYT_Nr?
```

This query might return `:WFMOUTPRE:BYT_NR 2`, indicating that there are 2 bytes per outgoing waveform data point.

Example 2

```
WFMOutpre:BYT_Nr 1
```

This command sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

WFMOutpre:NR_Pt?

Description

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

Related Commands

CURVe? (see page 467), DATa:SOUrce (see page 472), DATa:STARt (see page 473), DATa:STOP (see page 474)

Syntax

```
WFMOutpre:NR_Pt?
```

Example

```
WFMOutpre:NR_Pt?
```

This query might return `:WFMOUTPRE:NR_PT 5000`, indicating that there are 5000 data points to be sent.

WFMOutpre:PT_Fmt?

Description

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

Related Commands

CURVe? (see page 467), DATa:SOUrce (see page 472)

Syntax

```
WFMOutpre:PT_Fmt?
```

Example

```
WFMOutpre:PT_Fmt?
```

This query might return `:WFMOutpre:PT_Fmt ENV`, indicating that the waveform data are a series of max-min pairs.

WFMOutpre:PT_Off?

Description

This query-only command returns the trigger point relative to DATa:STARt for the waveform specified by the DATa:SOUrce command.

Note: This value is the point immediately following the actual trigger.

Group

Waveform Transfer

Related Commands

DATa:SOUrce (see page 472), DATa:STARt (see page 473), WFMOutpre:XZEro? (see page 506)

Syntax

```
WFMOutpre:PT_Off?
```

Example

```
WFMOutpre:PT_Off?
```

This query might return `:WFMOUTPRE:PT_OFF 251`, specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:PT_OR?

Description

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 200 (vertical) by 500 (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 500 possible horizontal columns being transmitted contains 200 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

Related Commands

DATa:SOUrce (see page 472)

Syntax

```
WFMOutpre:PT_OR?
```

Example 1

```
WFMOutpre:PT_OR?
```

This query might return `:WFMOUTPRE:PT_OR COL`, specifying that the waveform designated by the DATa:SOUrce waveform is a Fast Acquisition waveform.

Example 2

```
WFMOutpre:PT_OR?
```

This query might return `:WFMOUTPRE:PT_OR LINEAR` , specifying that the source pointed to is a non-Fast Acquisition waveform.

WFMOutpre:WFId?

Description

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

Related Commands

DATa:SOUrce (see page 472)

Syntax

```
WFMOutpre:WFId?
```

Outputs

<string> comprises the following comma-separated fields:

Waveform Suffixes		
Field	**Description**	**Examples**
Source	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 Source 1 waveform in the case of Dual Waveform Math	"AC coupling" "DC coupling" "GND coupling"
Vert Scale	A string giving the vertical scale factor of the unzoomed waveform. The numeric portion will always be 4 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 uVs/div" "10.00 kV/s/div" "200.0 mV/div" "50.00 unk/div"
Horiz Scale	A string giving the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four 4 digits. The examples cover all known internal units.	"100.0 ms/div" "10.00 kHz/div" "50.00 c/div"
Record Length	A string stating 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"
Primary Reference Offset	A string specifying the delta between the Primary Reference (typically, the A trigger) and the CURVe? zero reference location identified by a combination of PT_Off and XZEro in units of XUNits. For example, in Triggerable After Delay, this number would be the actual time between the A and B trigger. In the event that this number is not meaningful, the string will be exactly "0".	"57.2345 ms" "87.3 Hz" "0"

Example

```
WFMOutpre:WFId?
```

This query might return `:WFMOUTPRE:WFID "Ch1, DC coupling, 100.0mVolts/div,500.0µs/div,500 points, Hi Res mode"`

WFMOutpre:XINcr?

Description

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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:XUNit? (see page 507)

Syntax

```
WFMOutpre:XINcr?
```

Example

```
WFMOutpre:XINcr?
```

This query might return `:WFMOUTPRE:XINCR 10.0000E-6`, indicating that the horizontal sampling interval was 10 µs/point (500 µs/div).

WFMOutpre:XZEro?

Description

This query-only command returns the (subsample) 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 *stopped state operation* (i.e., ACQuire:STATE OFF), this is the only preamble that changes on each acquisition. If a query is run during *steady state operation* (i.e., 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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:XUNit? (see page 507)

Syntax

```
WFMOutpre:XZEro?
```

Example

```
WFMOutpre:XZEro?
```

This query might return `:WFMOUTPRE:XZERO 5.6300E-9`, indicating that the trigger actually occurred 5.63 ns before the trigger sample.

WFMOutpre:XUNit?

Description

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

Related Commands

DATa:SOUrce (see page 472)

Syntax

```
WFMOutpre:XUNit?
```

Example

```
WFMOutpre:XUNit?
```

This query might return `:WFMOUTPRE:XUNIT "Hz"`, indicating that the horizontal units for the waveform are measured in hertz.

WFMOutpre:YMUlt?

Description

This query-only command returns the vertical scale factor per digitizing level in units specified by the WFMOutpre:YUNit command for the waveform specified by the DATa:SOUrce command. For those formats in which WFMOutpre:BYT_Nr is important (all nonfloating 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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:BYT_Nr (see page 498), WFMOutpre:YUNit? (see page 510)

Syntax

```
WFMOutpre:YMUlt?
```

Example

```
WFMOutpre:YMUlt?
```

This query might return `:WFMOUTPRE:YMULT 4.0000E-3`, indicating that the vertical scale for the corresponding waveform was 100 mV/div.

WFMOutpre:YOFf?

Description

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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:BYT_Nr (see page 498)

Syntax

```
WFMOutpre:YOFf?
```

Example

```
WFMOutpre:YOFf?
```

This query 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?

Description

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

Related Commands

DATa:SOUrce (see page 472)

Syntax

```
WFMOutpre:YUNit?
```

Example

```
WFMOutpre:YUNit?
```

This query might return `:WFMOUTPRE:YUNIT "dB"`, indicating that the vertical units for the waveform are measured in decibels.

WFMOutpre:YZEro?

Description

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

Related Commands

DATa:SOUrce (see page 472), WFMOutpre:YUNit? (see page 510)

Syntax

```
WFMOutpre:YZEro?
```

Example

```
WFMOutpre:YUNit?
```

This query might return `:WFMOUTPRE:YZERO -100.0000E-3`, indicating that vertical offset was set to -100 mV.

Zoom Command Group

Zoom Overview

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

Zoom Commands

Command	Description
ZOOm?	Returns the current vertical and horizontal positioning and scaling of the display
ZOOm	Sets the zoom transforms to default values for either live traces or all traces
ZOOm:MODE?	Returns the current zoom mode
ZOOm:MODE	Turns zoom mode on or off
ZOOm:HORIzontal:LOCk?	Returns the waveforms that the horizontal zoom parameters affect
ZOOm:HORIzontal:LOCK	Sets the waveforms that the horizontal zoom parameters affect
ZOOm:<wfm>:HORIzontal:POSition?	Returns the zoom horizontal position of the specified waveform
ZOOm:<wfm>:HORIzontal:POSition	Sets the zoom horizontal position of the specified waveform
ZOOm:<wfm>:HORIzontal:SCAle?	Returns the zoom horizontal scale factor of the specified waveform
ZOOm:<wfm>:HORIzontal:SCAle	Sets the zoom horizontal scale factor of the specified waveform
ZOOm:<wfm>:VERTical:POSition?	Returns the zoom vertical position of the specified waveform
ZOOm:<wfm>:VERTical:POSition	Sets the zoom vertical position of the specified waveform
ZOOm:<wfm>:VERTical:SCAle?	Returns the zoom vertical scale of the specified waveform
ZOOm:<wfm>:VERTical:SCAle	Sets the zoom vertical scale of the specified waveform

ZOOm

Description

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 1

```
ZOOm {RESET|RESETLive}
```

Syntax 2

```
ZOOm?
```

Arguments

- `RESET`

  This resets the zoom transforms to default values for all traces.

- `RESETLive`

  This resets the zoom transforms to default values for live traces.

Example 1

```
ZOOm?
```

This query might return :

```
:ZOOM:MODE 0;GRATICULE:SIZE 50;:ZOOM:HORIZONTAL:LOCK ALL;
:ZOOM:CH1:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:CH1:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:CH2:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:CH2:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:CH3:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:CH3:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:CH4:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:CH4:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH1:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH1:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH2:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH2:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH3:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH3:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:MATH4:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:MATH4:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:REF1:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:REF1:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:REF2:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:REF2:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:REF3:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:REF3:VERTICAL:POSITION 0.0000;SCALE 1.0000;
:ZOOM:REF4:HORIZONTAL:POSITION 50.0000;SCALE 2;
:ZOOM:REF4:VERTICAL:POSITION 0.0000;SCALE 1.0000
```

Example 2

```
ZOOm RESET
```

This command resets the zoom transforms to default values for all traces.

ZOOm:MODe

Description

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 1

```
ZOOm:MODe {ON|OFF|<NR1>}
```

Syntax 2

```
ZOOm:MODe
```

Arguments

- `ON`

  This turns on Zoom mode.

- `OFF`

  This turns off Zoom mode.

- `<NR1>`

  A 0 turns off Zoom mode; any other value turns on Zoom mode.

Example 1

```
ZOOm:MODe?
```

This query might return `:WFMOUTPRE:ZOOM:MODE 1`, indicating that Zoom mode is currently turned on.

Example 2

```
ZOOm:MODe OFF
```

This command turns off Zoom mode.

ZOOm:HORizontal:LOCk

Description

This command sets or queries the waveforms that the horizontal zoom parameter affects. This is equivalent to setting **Horizontal Lock** in the Zoom side menu.

Group

Zoom

Syntax 1

```
ZOOm:HORizontal:LOCk {ALL|LIVe|NONe}
```

Syntax 2

```
ZOOm:HORizontal:LOCk?
```

Arguments

- `ALL`

  This specifies that all the (CH<x>, Ref<x>, Math<x>) waveforms will be horizontally positioned and scaled together.

- `LIVe`

  This specifies that all live (CH<x>) waveforms will be horizontally positioned and scaled together.

- `NONe`

  This specifies that only the selected waveform is positioned and scaled using the horizontal zoom parameters.

Example 1

```
ZOOm:HORizontal:LOCk?
```

This query might return `:ZOOM:HORIZONTAL:LOCK ALL`, indicating that all waveforms are positioned and scaled together.

Example 2

```
ZOOm:HORizontal:LOCk LIVE
```

This command horizontally positions and scales all live waveforms.

ZOOm:<wfm>:HORizontal:POSition

Description

This command sets or queries the horizontal position of specified waveform. The setting of the ZOOm:HORizontal:LOCk command determines the waveforms that are affected. For example, if ZOOm:HORizontal:LOCk is set to LIVe, then only live (as opposed to reference) waveforms are affected.

Group

Zoom

Syntax 1

```
ZOOm:<wfm>:HORizontal:POSition <NR3>
```

Syntax 2

```
ZOOm:<wfm>:HORizontal:POSition?
```

Argument

- `<NR3>`

  This is a value from 0 to 100 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 1x or greater.

Example 1

```
ZOOm:CH1:HORizontal:POSition?
```

This query might return `:ZOOM:CH1:HORIZONTAL:POSITION 50.0000`, indicating that the Zoom reference pointer for channel 1 is currently set at 50% of acquired waveform.

Example 2

```
ZOOm:CH1:HORizontal:POSition 50
```

This command sets the Zoom reference pointer for channel 1 at 50% of acquired waveform.

ZOOm:<wfm>:HORizontal:SCAle

Description

This command sets or queries the zoom horizontal scale factor of the specified waveform.

Group

Zoom

Syntax 1

```
ZOOm:<wfm>:HORizontal:SCAle <NR3>
```

Syntax 2

```
ZOOm:<wfm>:HORizontal:SCAle?
```

Argument

- `<NR3>`

  This is the amount of expansion in the horizontal direction.

Example 1

```
ZOOm:CH2:HORizontal:SCAle?
```

This query might return `:ZOOM:CH2:HORIZONTAL:SCALE 1`, indicating channel 2 has a horizontal scale factor of 1.

Example 2

```
ZOOm:CH1:HORizontal:SCAle 5
```

This command sets the channel 1 horizontal scale factor to 5.

ZOOm:<wfm>:VERTical:POSition

Description

This command sets or queries the Zoom vertical position of the specified waveform.

Group

Zoom

Syntax 1

```
ZOOm:<wfm>:VERTical:POSition <NR3>
```

Syntax 2

```
ZOOm:<wfm>:VERTical:POSition?
```

Argument

- `<NR3>`

  This is the vertical position, expressed in divisions.

Example 1

```
ZOOm:CH1:VERTical:POSition?
```

This query might return `:ZOOm:CH1:VERTical:POSition 0.0000`, indicating that the zoomed trace is centered at division 0 of the acquire trace.

Example 2

```
ZOOm:CH1:VERTical:POSition 2
```

This command sets the vertical position to 2, which centers the zoom trace at the second division of the acquire trace.

ZOOm:<wfm>:VERTical:SCAle

Description

This command sets or queries the zoom vertical scale of the specified waveform.

Group

Zoom

Syntax 1

```
ZOOm:<wfm>:VERTical:SCAle <NR3>
```

Syntax 2

```
ZOOm:<wfm>:VERTical:SCAle?
```

Argument

- `<NR3>`

  This 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).

Example 1

```
ZOOm:CH2:VERTical:SCAle?
```

This query might return `:ZOOM:CH2:VERTICAL:SCALE 2.0000`, indicating that the vertical scale is 2x.

Example 2

```
ZOOm:REF1:VERTical:SCAle 4
```

This command sets the vertical scale of REF1 to 5x.

Status and Events

The oscilloscope provides a status and event reporting system for the GPIB interface. This system informs you of certain significant events that occur within the oscilloscope.

The oscilloscope status handling system consists of five 8-bit registers and two queues. 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 oscilloscope. They include the Standard Event Status Register (SESR) and the Status Byte Register (SBR).
- 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 oscilloscope 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 oscilloscope 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 oscilloscope. 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.

Standard Event Status Register (SESR)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

SESR Bit Functions

Bit	Function	
7 (MSB)	PON	Power On. Shows that the oscilloscope was powered on. On completion, the diagnostic tests also set this bit.
6	URQ	User Request. Shows that an Application menu button was pressed.
5	CME	Command Error. Shows that an error occurred while the oscilloscope was parsing a command or query.
4	EXE	Execution Error. Shows that an error occurred while the oscilloscope was parsing a command or query.
3	DDE	Device Error. Shows that a device error occurred.
2	QYE	Query Error. Shows that 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	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 oscilloscope 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.

Status Byte Register (SBR)

	6						
7	RQS	5	4	3	2	1	0
----	6	ESB	MAV	----	----	----	----
	MSS						

SBR Bit Functions

Bit	Function	
7 (MSB)	-------	Not used.
6	RQS	Request Service. Obtained from a serial poll. Shows that the oscilloscope 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.
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 the 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.

Device Event Status Enable Register (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.

Event Status Enable Register (ESER)

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.

Service Request Enable Register (SRER)

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: 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

Overview

The oscilloscope status and event reporting system contains two queues: the Output Queue and the Event Queue.

Output Queue

The oscilloscope 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 oscilloscope 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.

Messages

Overview

The information contained in the topic tabs above covers all the programming interface messages the oscilloscope generates in response to commands and queries.

For most messages, a secondary message from the oscilloscope 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.

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.

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.

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, Zero period
2202	Measurement error, No period found
2203	Measurement error, No period, second waveform
2204	Measurement error, Low signal amplitude
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, Waveform does not cross Mid Ref

2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No 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 meas cannot be performed on this type of wfm
2244	Source wfm 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 wfm 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
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 wfm

Device Error

The following table lists the device errors that can occur during oscilloscope operation. These errors may indicate that the oscilloscope needs repair.

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.

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.

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 in minmax
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

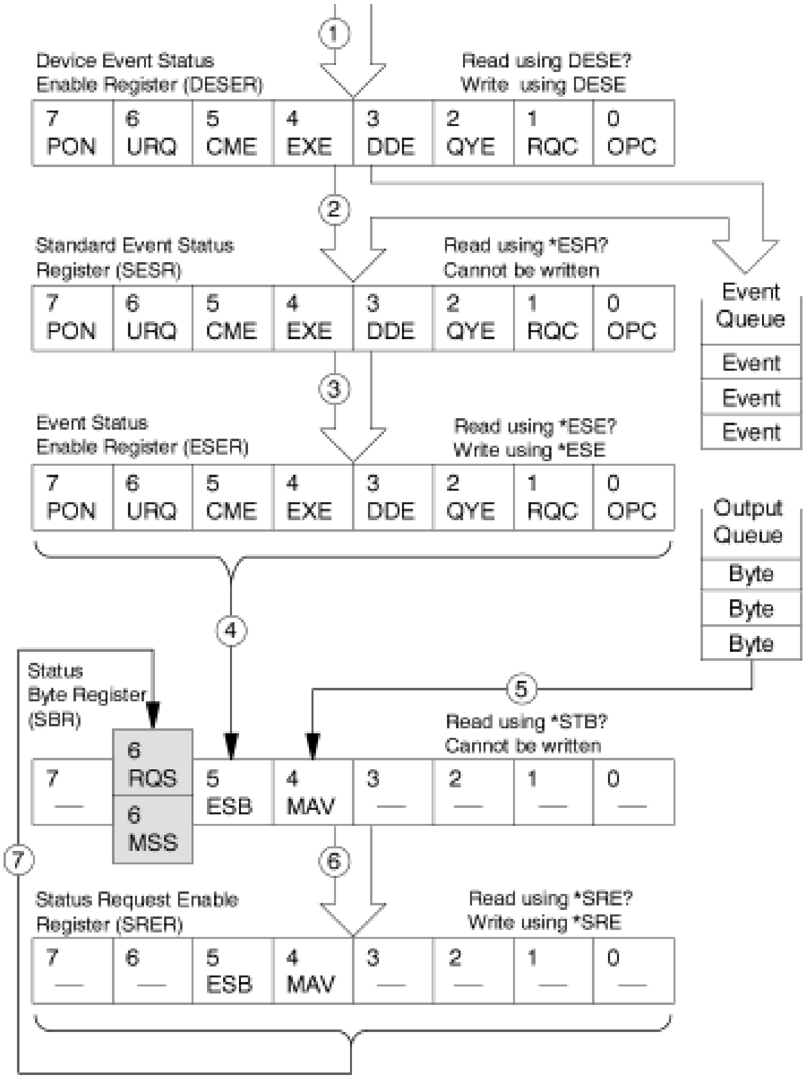
The following table shows internal errors that indicate an internal fault in the oscilloscope.

Internal Warning Messages

Code	Message
600	Internal warning
630	Internal warning, 50 ohm overload

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.



Device Event Status Enable Register (DESER)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Standard Event Status Register (SESR)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Event Status Enable Register (ESER)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Status Byte Register (SBR)

7	6	5	4	3	2	1	0
—	RQS 6 MSS	ESB	MAV	—	—	—	—

Status Request Enable Register (SRER)

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

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). Bit 2 of the SBR (EAV) flags the presence of events in the queue.

Synchronization Methods

Overview

Although most GPIB commands are completed almost immediately after being received by the oscilloscope, some commands start a process that requires more time. For example, once a single sequence acquisition command is executed, it may be a few seconds before the hardcopy operation is complete. Rather than remain idle while the operation is in process, the oscilloscope 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 oscilloscope 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  500
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 oscilloscope takes an amplitude measurement (see figure below). This can result in an incorrect amplitude value.

```
    ___________________
    |ACQUIRE:STATE  ON  |
    +-----------------+

                                           ____________
                                           |Acquiring  |
                                           |Waveform  |
                                           |Data           |
                                           +----------+

___________________
                                                           |Take
Measurement  |
                                                           +--------------
---+
|<---------------------------------------------------->|
                                          Processing  Time
```

```
===========================================================
```

To ensure the oscilloscope completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program.

```
===========================================================
___________________
|ACQUIRE:STATE  ON  |
+-----------------+_____________
                                      |Acquiring    |
                                      |Waveform      |
                                      |Data              |
                                      +-----------
+___________________________
                                                                |Take
Measurement                  |
                                                                +------------
-------------+
|<-------------------------------------------------------->|
                                            Processing  Time

===========================================================
```

You can use four commands to synchronize the operation of the oscilloscope with your application program: *WAI, BUSY?, *OPC, and *OPC?

*WAI

You can force commands to execute sequentially by using the *WAI command. This command forces completion of the previous commands before processing new ones.

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 500
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?
```

Though *WAI is one of the easiest ways to achieve synchronization, it is also the most costly. The processing time of the oscilloscope is slowed since it is processing a single command at a time. This time could be spent doing other tasks.

The controller can continue to write commands to the input buffer of the oscilloscope, but the commands will not be processed by the oscilloscope until all 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 oscilloscope 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  500
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  500
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  500
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
```

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 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 500
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.

Programming Examples

Overview

The example programs illustrate methods you can use to control the oscilloscope from the GPIB interface. These programs are listed on your product CD and are auto-installed in an example directory for you.

The programs run on a PC compatible system equipped with a National Instruments GPIB board and associated drivers. For example, the GPIB programs work with a National Instruments AT-GPIB/TNT GPIB package (see Figure 4-1).

Compiling the Example GPIB Programs

All the example GPIB programs assume that the GPIB system recognizes the oscilloscope as DEV1 and the PC (controller) as GPIB0. You can assign these names using different methods depending on your operating system. In Windows NT, you can assign these names through the Control Panel→GPIB→Configure Button. In Windows 95/98, you can assign the GPIB name through the Control Panel→System Icon→Device Manager→Display→AT-GPIB/TNT→GPIB.

The example GPIB software includes:

MEAS: measures a parameter of an oscilloscope waveform.

TL: a talker/listener program.

The example programs diskette contains programs written in Microsoft Visual C^{++} 6.0.

Visual C^{++} code is in V-CPLUS6. Within this directory, an executable version of the program is in the PROGRAMS subdirectory. Source versions are in the SOURCES subdirectory.

A README file in each directory has the latest documentation.

If you wish to develop code, you will need to use files that come with the GPIB system. Specifically, the Visual C^{++} programs use DECL_32.h and GPIB_32.obj.

Note: The GPIB programs you compile in the Sources directory work with the National Instruments AT-GPIB/TNT GPIB system. It may take extra steps or changes to get them to work with older Tektronix GURU and other GPIB systems.

Compiling and Linking Example Visual C++ Programs

To make an executable for any of the files (for example, AUTOSET project), perform the following steps:

1 Install Visual C^{++}.

7 Install the National Instruments AT-GPIB/TNT GPIB board and drivers. *Remember to identify the GPIB device as DEV1* (see GPIB Examples section for more information) *and to reboot your PC to initialize the GPIB drivers*.

8 Copy the files from the examples diskette to your hard disk. For example, if the current drive is hard disk C, you want to store the examples in drive C, and the examples diskette in drive B:
a Invoke the Microsoft Explorer program.
b Drag the b:\GPIB\V-CPLUS6\SOURCES folder onto the C drive.

9 Set up a project for each example. The example below creates a new project for the AUTOSET example program.
a Invoke Visual C^{++}.
b Select "New" from the File menu.
c Choose "New Win32 Console Application".
d Give the project a name, for example, GeTwfm.
e Select "Empty Project".
f Add the following files to the project from the Project menu:

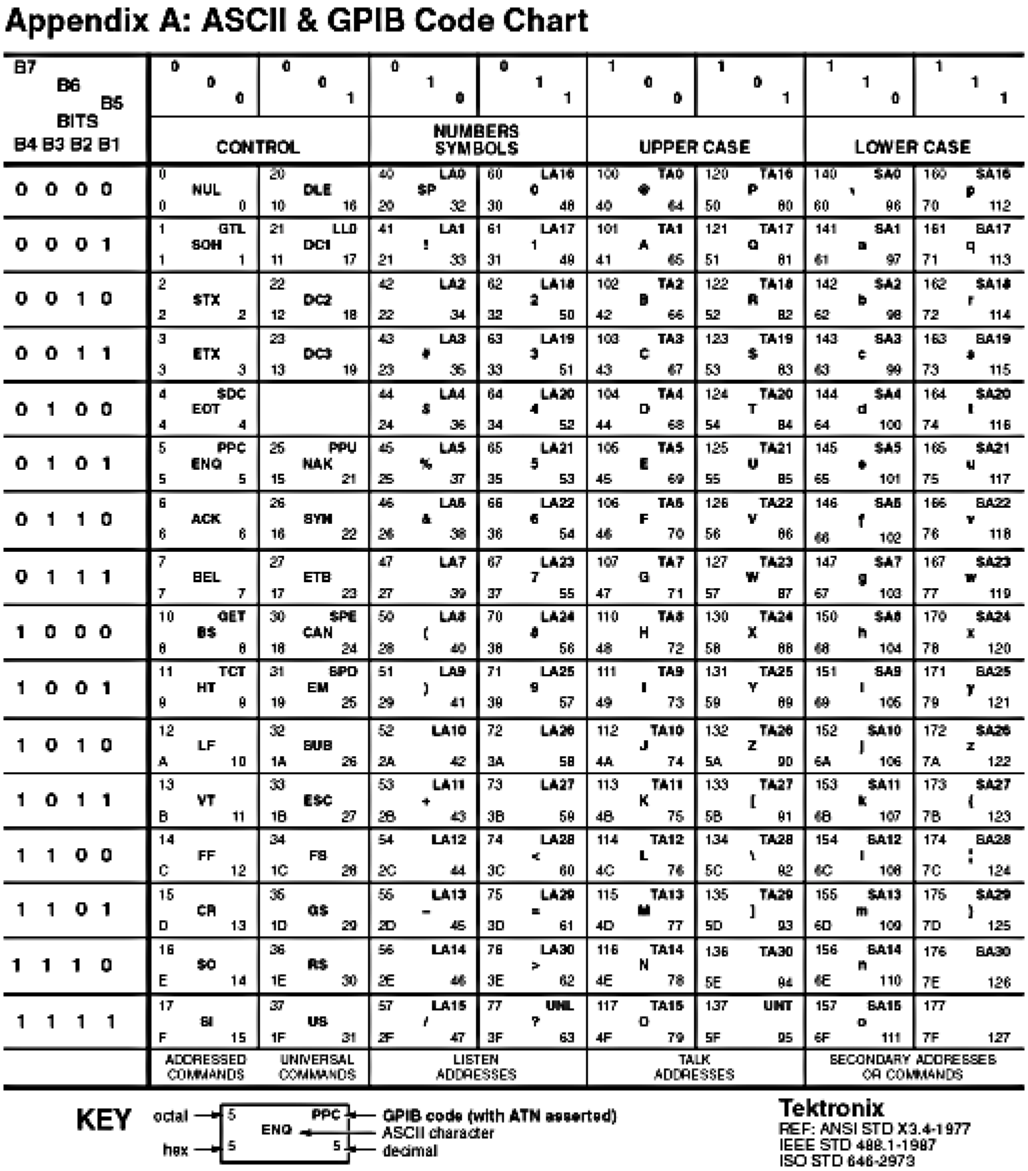
getwfm.c
gpibwrite.c
gpiberr.c
gpibread.c
gpibwait.c
decl-32.h (provided by National Instruments)
gpib-32.obj (provided by National Instruments)

10 To compile and link your sample program, choose "BUILD ALL" from the Build menu.

11 To run the sample program, choose RUN from the "BUILD" menu.

Miscellaneous

Character Charts



Appendix A: ASCII & GPIB Code Chart

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		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 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY

octal → 5 PPC ← GPIB code (with ATN asserted)

ENQ ← ASCII character

hex → 5 5 ← decimal

Tektronix

REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Reserved Words

Overview

This topic is a list of reserved words for the TDS7000 series oscilloscopes. Capital letters identify the required minimum spelling. For the most robust code use the full spelling, since spelling rules may change over time and from instrument model to instrument model.

*CAL to DIAg

*CAL
*CLS
*DDT
*ESE
*ESR
*IDN
*LRN
*OPC
*OPT
*PSC
*PUD
*RCL
*RST
*SAV
*SDS
*SRE
*STB
*TRG
*TST
*WAI
A
ABSolute
ACQuire
ACTUal
ALIas
ALL
ALLev
AREA
AUTOBright
AUTOSet
AUXout
B
BANdwidth
BELl
BETWeen
BIT_Nr
BN_Fmt
BOTh
Box
BOXPcnt
BUSY
BY
BYT_Nr
BYT_Or
CALibrate
CENTER
CH<x>
CLAss
CLOCk
CMDBatch
COLOr
CONTROL
COPy
COUNt
COUPling
CURSor
CURVe
DATa
DATe
DEFIne
DELay
DELEte
DELTa
DELTATime
DESE
DESKew
DESTination
DIAg

DIR to MAXimum

DIR
DIREction
DISplay
DIVisions
EDGE
EDGE2
ENCdg
ERRLOG
EVENT
EVENTS
EVMsg
EVQty
EXECUTE
EXTatten
EXTDBatten
EXTUnits
FACtory
FAILURES
FASTAcq
FILEName
FILESystem
FILTer
FIRST
FORMat
FRAME
FREQuency
FUNCtion
GAIN
GATEPOS
GATEWIDTH
GATING
GLItch
GPIB
GRATicule
HALT
HARDCopy
HBArs
HDELTA
HDR
HEADer
HIGH
HIGHLimit
HIStogram
HOLDoff
HOLDTime
HORizontal
HPOS<x>
ID
IMMed
INTENSITy
INPut
ITEM
LABEL
LAST
LENgth
LESSLimit
LEVEL
LOCk
LOGIc
LOOP
LOOPS
LOW
LOWLimit
MAG
MAIn
MATH<x>
MATHCOLOr
MAXimum

MEAN to SIZe

MEAN
MEAS<x>
MEASUrement
METHod
MID
MID2
MINimum
MKDir
MODe
MORELimit
NAMe
NEWpass
NEXT
NR_Pt
NUMACq
NUMAVg
NUMEnv
NUMITEMS
OFFSet
PAIred
PALETTE
PASSword
PATtern
PERCent
PERSistence
PHASE
POLarity
PORT
POSition
POSition<x>
PRInt
PRObe
PROBECal
PROBEFunc
PRObestate
PT_Fmt
PT_Off
PT_OR
PULse
PUMODE
READFile
RECAll
RECOrdlength
REF
REF<x>
REFCOLOr
REFLevel
REName
REPEt
RESBw
RESistance
RESULT
RESults
RMDir
RUNT
SAMPLERate
SAVe
SCAle
SCREENSAVER
SCREENSAVERDELAY
SELect
SELECTED
SERnumber
SET
SETHold
SETTime
SETUp
SIZe

SLOpe to ZOOm

SLOpe
SOURCE
SOURCE2
SPAN
SPC
SPECTral
SPLit
STARt
STATE
STATIstics
STDdev
STOP
STOPAfter
STRing
STYLe
SUBITEMS
SUBSYS
SUPPress
TERminator
TEST
THReshold
TIMe
TIMEOut
TIMEStamp
TRACk
TRANsition
TRIGBar
TRIGger
TRIGIF
TRIGT
TYPe
UNIts
UNLock
UNWRap
VALue
VARPersist
VBArs
VDELTA
VERBose
VOLtage
<wfm>
WAVEform
WAVFrm
WEIghting
WFId
WFMInpre
WFMOutpre
WHEn
WIDth
WINdow
WRITEFile
XINcr
XPOS
XUNit
XZEro
YMUlt
YOFf
YPOS
YUNit
YZEro
ZOOm

Factory Default Setup Values

Overview

This topic contains tables that list the default setup values by command group. These are the values that the instrument sets when you press the DEFAULT SETUP front-panel button or send the FACtory command. Only those commands that have values set by the DEFAULT SETUP function are listed in these tables.

Acquisition

Default Setup Values	
Command	**Default**
ACQuire:MODe	SAMple
ACQuire:NUMAVg	16
ACQuire:NUMEnv	10
ACQuire:REPEt	1 (ON)
ACQuire:STATE	1 (ON)
ACQuire:STOPAfter	RUNSTop
FASTAcq:STATE	0 (OFF)

Cursor

Default Setup Values	
Command	**Default**
CURSor:FUNCtion	VBArs
CURSor:HBArs:POSITION<x>	±300.0000E-9
CURSor:MODe	INDependent
CURSor:PAIred:POSITION<x>	±800.0000E-9
CURSor:SOUrce	CH1
CURSor:SPLit:POSITION<x>	±800.0000E-9
CURSor:SPLit:SOURCE2	CH1
CURSor:SPLit:UNIts	BASE
CURSor:STATE	OFF
CURSor:VBArs:POSITION<x>	±800.0000E-9
CURSor:VBArs:UNIts	SECOnds

Display

Default Setup Values	
Command	**Default**
DISplay:CLOCk	1
DISplay:COLOr:MATHCOLOr	DEFAULT
DISplay:COLOr:PALETTE	NORMal
DISplay:COLOr:REFCOLOr	DEFAULT
DISplay:FILTer	SINX
DISplay:FORMat	YT
DISplay:INTENSITy:AUTOBright	1 (ON)
DISplay:INTENSITy:SCREENSAVER	1 (ON)
DISplay:INTENSITy:SCREENSAVERDELAY	28800
DISplay:INTENSITy:WAVEform	75.0000
DISplay:PERSistence	OFF
DISplay:STYle	VECtors
DISplay:TRIGBar	SHORt
DISplay:TRIGT	1 (ON)
DISplay:VARPersist	500.0000E-3

Hardcopy

Default Setup Values	
Command	**Default**
HARDCopy:FILEName	""
HARDCopy:PORT	FILE

Histogram

Default Setup Values	
Command	**Default**
HIStogram:BOXPcnt	30.0000, 25.1000, 70.0000, 75.2000
HIStogram:DISplay	0 (OFF)
HIStogram:FUNCTION	HORizontal
HIStogram:SIZe	2.0000
HIStogram:SOUrce	CH1
HIStogram:STATE	0 (OFF)

Horizontal

Default Setup Values

Command	Default
HORizontal:FASTframe:COUNt	2
HORizontal:FASTframe:LENgth	5000
HORizontal:FASTframe:REF:FRAme	1
HORizontal:FASTframe:REF:SOUrce	CH1
HORizontal:FASTframe:SELECTED:<wfm>	CH1-4 2 MATH1-4 2 REF1-4 2
HORizontal:FASTframe:STATE	0
HORizontal:FASTframe:TRACk	LIVE
HORizontal:RECOrdlength	5000
HORizontal:RESOlution	5000
HORizontal:ROLL	AUTO
HORizontal:SPECTral:RECOrdlength	5000
HORizontal[:MAIn]:DELay:MODe	0 (OFF)
HORizontal[:MAIn]:DELay:POSition	50.0000
HORizontal[:MAIn]:DELay:TIMe	0.0000
HORizontal[:MAIn]:POSition	50.0000
HORizontal[:MAIn]:SAMPLERate	2.5000E+9
HORizontal[:MAIn]:SCAle	200.0000E-9

Math

Default Setup Values	
Command	**Default**
MATH<x>:DEFine	" "
MATH<x>:LABEL:NAMe	" "
MATH<x>:LABEL:XPOS	5
MATH<x>:LABEL:YPOS	MATH1 65 MATH2 80 MATH3 95 MATH4 110
MATH<x>:NUMavg	2
MATH<x>:POSition	0.0000
MATH<x>:SCAle	1.0000
MATH<X>:SPECTral:CENTER	625.0000E+06
MATH<X>:SPECTral:GATEPOS	0.0000
MATH<X>:SPECTral:GATEWIDTH	2.0000E-06
MATH<X>:SPECTral:LOCk	0 (OFF)
MATH<X>:SPECTral:MAG	DB
MATH<X>:SPECTral:PHASE	DEGREES
MATH<X>:SPECTral:REFLEVEL	20.0000
MATH<X>:SPECTral:REFLEVELOffset	223.6000E-03
MATH<X>:SPECTral:RESBw	1.0000E+06
MATH<X>:SPECTral:SPAN	1.2500E+09
MATH<X>:SPECTral:SUPPress	-35.0000
MATH<X>:SPECTral:UNWRap	0 (OFF)
MATH<X>:SPECTral:WINdow	GAUSSIAN

Measurement

Default Setup Values	
Command	**Default**
MEASUrement:GATING	0 (OFF)
MEASUrement:METHod	HIStogram
MEASUrement:IMMed:TYPe	undefined
MEASUrement:IMMed:SOURCE[1]	CH1
MEASUrement:IMMed:SOURCE2	CH1
MEASUrement:IMMed:DELay:EDGE[1]	RISe
MEASUrement:IMMed:DELay:EDGE2	RISe
MEASUrement:IMMed:DELay:DIREction	FORWards
MEASUrement: MEAS<x>:STATE	0 (OFF)
MEASUrement:MEAS<x>:SOURCE[1]	CH1
MEASUrement:MEAS<x>:SOURCE2	CH1
MEASUrement:MEAS<x>:DELay:EDGE[1]	RISe
MEASUrement:MEAS<x>:DELay:EDGE2	RISe
MEASUrement:MEAS<x>:DELay:DIREction	FORWards
MEASUrement:REFLevel:METHod	Percent
MEASUrement:REFLevel:ABSolute:HIGH	0.0000
MEASUrement:REFLevel:ABSolute:LOW	0.0000
MEASUrement:REFLevel:ABSolute:MID	0.0000
MEASUrement:REFLevel:ABSolute:MID2	0.0000
MEASUrement:REFLevel:PERCent:HIGH	90.0000
MEASUrement:REFLevel:PERCent:LOW	10.0000
MEASUrement:REFLevel:PERCent:MID	50.0000
MEASUrement:REFLevel:PERCent:MID2	50.0000
MEASUrement:STATIstics:MODe	OFF
MEASUrement:STATIstics:WEIghting	32

Miscellaneous

Default Setup Values	
Command	**Default**
AUXout:EDGe	FALling
AUXout:SOUrce	ATRIGger
CMDBatch	1 (ON)
HEADer	1 (ON)
LOCk	NONe
VERBose	1 (ON)

Trigger

Default Setup Values

Command	Default
TRIGger:A:MODe	AUTO
TRIGger:A:TYPe	EDGE
TRIGger:A:LEVel	0.0000
TRIGger:A:HOLDoff:BY	DEFAult
TRIGger:A:HOLDoff:TIMe	250.0000E-09
TRIGger:A:EDGE:SOUrce	CH1
TRIGger:A:EDGE:COUPling	DC
TRIGger:A:EDGE:SLOpe	RISe
TRIGger:A:LOGIc:CLAss	PATtern
TRIGger:A:LOGIc:FUNCtion	AND
TRIGger:A:LOGIc:THReshold:CH<x>	CH1 1.2000 CH2 1.2000 CH3 1.2000 CH4 1.2000
TRIGger:A:LOGIc:INPut:CH<x>	CH1 HIGH CH2 X CH3 X
TRIGger:A:LOGIc:PATtern:INPut:CH4	X
TRIGger:A:LOGIc:PATtern:WHEn	TRUE
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit	5.0000E-09
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit	5.0000E-09
TRIGger:A:LOGIc:SETHold:CLOCk:EDGE	RISe
TRIGger:A:LOGIc:SETHold:CLOCk:THReshold	1.2000
TRIGger:A:LOGIc:SETHold:CLOCk:SOUrce	CH2
TRIGger:A:LOGIc:SETHold:DATa:THReshold	1.2000
TRIGger:A:LOGIc:SETHold:DATa:SOUrce	CH1
TRIGger:A:LOGIc:SETHold:HOLDTime	2.0000E-09
TRIGger:A:LOGIc:SETHold:SETTime	3.0000E-09
TRIGger:A:LOGIc:STATE:INPut:CH4	RISe
TRIGger:A:LOGIc:STATE:WHEn	TRUe
TRIGger:A:PULse:CLAss	GLItch
TRIGger:A:PULse:SOUrce	CH1
TRIGger:A:PULse:GLItch:WIDth	2.0000E-09
TRIGger:A:PULse:GLItch:TRIGIF	ACCept
TRIGger:A:PULse:GLItch:POLarity	POSITIVe
TRIGger:A:PULse:RUNT:POLarity	POSITIVe
TRIGger:A:PULse:RUNT:THReshold:HIGH	1.2000E-00
TRIGger:A:PULse:RUNT:THReshold:LOW	800.0000E-03
TRIGger:A:PULse:RUNT:WHEn	OCCurs
TRIGger:A:PULse:RUNT:WIDth	2.0000E-09
TRIGger:A:PULse:TRANsition:DELTATime	2.0000E-09
TRIGger:A:PULse:TRANsition:POLarity	POSITIVe
TRIGger:A:PULse:TRANsition:THReshold:HIGH	1.2000E-00
TRIGger:A:PULse:TRANsition:THReshold:LOW	800.0000E-03
TRIGger:A:PULse:TRANsition:WHEn	FASTERthan
TRIGger:A:PULse:WIDth:LOWLimit	2.0000E-09
TRIGger:A:PULse:WIDth:HIGHLimit	2.0000E-09
TRIGger:A:PULse:WIDth:WHEn	WITHIN
TRIGger:A:PULse:WIDth:POLarity	POSITIVe
TRIGger:A:PULse:TIMEOut:POLarity	STAYSHigh
TRIGger:A:PULse:TIMEOut:TIMe	2.0000E-09
TRIGger:B:STATE	0 (OFF)
TRIGger:B:TYPe	EDGe
TRIGger:B:LEVel	0.0000
TRIGger:B:BY	EVENTS
TRIGger:B:EDGE:SOUrce	CH1
TRIGger:B:EDGE:SLOpe	RISe
TRIGger:B:EDGE:COUPling	DC
TRIGger:B:TIMe	16.0000E-09
TRIGger:B:EVENTS:COUNt	2

Vertical

Default Setup Values	
Command	**Default**
CH<x>:BANdwidth	CH1-4 1.0000E+09
CH<x>:COUPling	CH1-4 DC
CH<x>:DESKew	CH1-4 0.0000
CH<x>:OFFSet	CH1-4 0.0000
CH<x>:POSition	CH1-4 0.0000
CH<x>:SCAle	CH1-4 100.0000E-03
CH<x>:TERmination	CH1-4 1.0000E-06
CH<x>:PROBEFunc:EXTatten	CH1-4 1.0000
CH<x>:PROBEFunc:EXTUnits	CH1-4 "None"
CH<x>:LABEL:NAMe	CH1-4 " "
CH<x>:LABEL:XPOS	CH1-4 5
CH<x>:LABEL:YPOS	CH1 5 CH2 20 CH3 35 CH4 50
CURVe	
DATa:DESTination	REF1
DATa:ENCdg	RIBinary
DATa:SOUrce	CH1
DATa:STARt	1
DATa:STOP	500
REF<x>:LABel:NAMe	" "
REF<x>:LABel:XPOS	REF1-4 5
REF<x>:LABel:YPOS	REF1 125 REF2 140 REF3 155 REF4 170
REF<x>:SCAle	1.0000
REF<x>:POSition	0.0000
SELect:CH<x>	CH1 1 CH2 0 CH3 0 CH4 0
SELect:CONTROl	CH1
SELect:MATH<x>	MATH1-4 0
SELect:REF<x>	REF1-4 0

Waveform

Default Setup Values	
Command	**Default**
WFid	"Ch1, DC coupling, 100.0mV/div, 2000.0ns/div, 5000 points, Sample mode"
WFMInpre:BIT_Nr	8
WFMInpre:BN_Fmt	RI
WFMInpre:BYT_Nr	1
WFMInpre:BYT_Or	MSB
WFMInpre:ENCdg	BIN
WFMInpre:NR_Pt	500
WFMInpre:PT_Fmt	Y
WFMInpre:PT_Off	0
WFMInpre:XINcr	2.0000E-06
WFMInpre:XUNit	"s"
WFMInpre:XZEro	0.000
WFMInpre:YMUlt	1.0000E-03
WFMInpre:YOff	0.0000
WFMInpre:YUNit	"V"
WFMInpre:YZEro	0.0000
WFMOutpre:BIT_Nr	8
WFMOutpre:BN_Fmt	RI
WFMOutpre:BYT_Nr	1
WFMOutpre:BYT_Or	MSB
WFMOutpre:ENCdg	BIN
WFMOutpre:NR_Pt	500
WFMOutpre:PT_Fmt	Y
WFMOutpre:PT_Off	2500
WFMOutpre:XINcr	400.0000E-12
WFMOutpre:XUNit	"s"
WFMOutpre:XZEro	0.000
WFMOutpre:YMUlt	4.0000E-03
WFMOutpre:YOff	0.0000
WFMOutpre:YUNit	"V"
WFMOutpre:YZEro	0.0000

Zoom

Default Setup Values	
Command	**Default**
ZOOm:MODe	0 (OFF)
ZOOm:HORizontal:LOCk	ALL
ZOOm:<wfm>:HORizontal:POSition	CH1-4 50.0000 MATH1-4 50.0000 REF1-4 50.0000
ZOOm:<wfm>:HORizontal:SCAle	CH1-4 2 MATH1-4 2 REF1-4 2
ZOOm:<wfm>:VERTical:POSition	CH1-4 0.0000 MATH1-4 0.0000 REF1-4 0.0000
ZOOm:<wfm>:VERTical:SCAle	CH1-4 1.0000 MATH1-4 1.0000 REF1-4 1.0000

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.

GPIB Functions

The following table lists the GPIB interface functions and electrical functions subsets supported by this instrument with brief descriptions of each.

GPIB Functions

Interface Function	Subset	Meaning
Acceptor Handshake	AH1	The instrument can receive multiline messages across the GPIB from other devices.
Controller	C0	No Controller capability; the instrument cannot control other devices.
Device Clear	DC1	The instrument can respond to both the DCL (Device Clear) interface message and to the Selected Device Clear (SDC) interface message when the instrument is listen-addressed.
Device Trigger	DT0	No Device Trigger capability; the instrument does not 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 its 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 its 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), PPD (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 the 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 GPIB to other devices.
Talker	T5	The instrument becomes a talker when it detects its 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 its listen address being sent over the bus with the ATN line asserted. The instrument also ceases to be a talker when it detects another device's talk address being sent over the data lines with ATN asserted.

Interface Messages

The following table shows the standard interface messages that the instrument supports.

Standard Interface Messages	
Message	**GPIB**
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