

# **Programmer Manual**



## **TDS 200-Series Digital Real-Time Oscilloscope 071-0493-01**

This document supports TDS 210 and TDS 220 with FV:v1.09 and above when used with TDS2CM version CMV:v1.04 and above, or TDS2MM any version, and TDS224 when used with any version of TDS2CM or TDS2MM.

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

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (c)(1) and (2) of the Commercial Computer Software – Restricted Rights clause at FAR 52.227-19, as applicable.

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

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

**THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.**



# Table of Contents

<b>General Safety Summary</b> .....	<b>v</b>
<b>Preface</b> .....	<b>vii</b>
Related Manuals .....	vii
Conventions .....	viii
Contacting Tektronix .....	viii

## Getting Started

<b>Getting Started</b> .....	<b>1-1</b>
------------------------------	------------

## Syntax and Commands

<b>Command Syntax</b> .....	<b>2-1</b>
Command and Query Structure .....	2-2
Command Entry .....	2-5
Constructed Mnemonics .....	2-8
Argument Types .....	2-9
<b>Command Groups</b> .....	<b>2-15</b>
Acquisition Commands .....	2-15
Calibration and Diagnostic Commands .....	2-16
Cursor Commands .....	2-17
Display Commands .....	2-18
Hard Copy Commands .....	2-18
Horizontal Commands .....	2-19
Measurement Commands .....	2-20
Miscellaneous Commands .....	2-21
RS-232 Commands .....	2-22
Save and Recall Commands .....	2-23
Status and Error Commands .....	2-23
Trigger Commands .....	2-24
Vertical Commands .....	2-25
Waveform Commands .....	2-26
<b>Command Descriptions</b> .....	<b>2-33</b>

**Status and Events**

<b>Status and Events .....</b>	<b>3-1</b>
Registers .....	3-1
Queues .....	3-6
Event Handling Sequence .....	3-8
Synchronization Methods .....	3-10
Messages .....	3-17

**Appendices**

<b>Appendix A: ASCII Code Chart .....</b>	<b>A-1</b>
<b>Appendix B: Factory Setup .....</b>	<b>B-1</b>

**Glossary and Index**

## List of Figures

Figure 2–1: Command message elements .....	2–3
Figure 2–2: Block Argument example .....	2–12
Figure 3–1: The Standard Event Status Register (SESR) ..	3–2
Figure 3–2: The Status Byte Register (SBR) .....	3–3
Figure 3–3: The Device Event Status Enable Register (DESER) .....	3–4
Figure 3–4: The Event Status Enable Register (ESER) ....	3–5
Figure 3–5: The Service Request Enable Register (SRER) .....	3–5
Figure 3–6: Status and Event Handling Process .....	3–9
Figure 3–7: Command Processing Without Using Synchronization .....	3–11
Figure 3–8: Processing Sequence With Synchronization ...	3–11

## List of Tables

Table 2–1: BNF notation .....	2–1
Table 2–2: Command message elements .....	2–2
Table 2–3: Comparison of header off and on responses ...	2–4
Table 2–4: Acquisition Commands .....	2–15
Table 2–5: Calibration and Diagnostic Commands .....	2–16
Table 2–6: Cursor Commands .....	2–17
Table 2–7: Display Commands .....	2–18
Table 2–8: Hard Copy Commands .....	2–18
Table 2–9: Horizontal Commands .....	2–19

<b>Table 2–10: Measurement Commands</b>	<b>2–20</b>
<b>Table 2–11: Miscellaneous Commands</b>	<b>2–21</b>
<b>Table 2–12: RS-232 Commands</b>	<b>2–22</b>
<b>Table 2–13: Save and Recall Commands</b>	<b>2–23</b>
<b>Table 2–14: Status and Error Commands</b>	<b>2–23</b>
<b>Table 2–15: Trigger Commands</b>	<b>2–24</b>
<b>Table 2–16: Vertical Commands</b>	<b>2–25</b>
<b>Table 2–17: Waveform Commands</b>	<b>2–26</b>
<b>Table 2–18: Binary data ranges</b>	<b>2–30</b>
<b>Table 2–19: Vertical position ranges using a 1X probe</b>	<b>2–50</b>
<b>Table 2–20: DATA and WFMPre Parameter Settings</b>	<b>2–68</b>
<b>Table 2–21: Commands that generate an Operation</b>	
Complete message	2–123
<b>Table 2–22: Additional WFMPre commands</b>	<b>2–175</b>
<b>Table 3–1: SESR Bit Functions</b>	<b>3–2</b>
<b>Table 3–2: SBR Bit Functions</b>	<b>3–3</b>
<b>Table 3–3: No Event Messages</b>	<b>3–17</b>
<b>Table 3–4: Command Error Messages - CME Bit 5</b>	<b>3–17</b>
<b>Table 3–5: Execution Error Messages - EXE Bit 4</b>	<b>3–18</b>
<b>Table 3–6: Device Error Messages - DDE Bit 3</b>	<b>3–21</b>
<b>Table 3–7: System Event Messages</b>	<b>3–21</b>
<b>Table 3–8: Execution Warning Messages - EXE Bit 4</b>	<b>3–22</b>
<b>Table 3–9: Internal Warning Messages</b>	<b>3–23</b>



# General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

*Only qualified personnel should perform service procedures.*

## Injury Precautions

**Use Proper Power Cord.** To avoid fire hazard, use only the power cord specified for this product.

**Avoid Electric Overload.** To avoid injury or fire hazard, do not apply potential to any input, including the common inputs, that varies from ground by more than the maximum rating for that input.

**Avoid Electric Shock.** To avoid injury or loss of life, do not connect or disconnect probes or test leads while they are connected to a voltage source.

**Do Not Operate Without Covers.** To avoid electric shock or fire hazard, do not operate this product with covers or panels removed.

**Do Not Operate in Wet/Damp Conditions.** To avoid electric shock, do not operate this product in wet or damp conditions.

**Do Not Operate in an Explosive Atmosphere.** To avoid injury or fire hazard, do not operate this product in an explosive atmosphere.

## Product Damage Precautions

**Use Proper Power Source.** Do not operate this product from a power source that applies more than the voltage specified.

**Do Not Operate With Suspected Failures.** If you suspect there is damage to this product, have it inspected by qualified service personnel.

## Safety Terms and Symbols

**Terms in This Manual.** These terms may appear in this manual:



---

**WARNING.** *Warning statements identify conditions or practices that could result in injury or loss of life.*

---



---

**CAUTION.** *Caution statements identify conditions or practices that could result in damage to this product or other property.*

---

**Terms on the Product.** These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

**Symbols on the Product.** These symbols may appear on the product:



DANGER  
High Voltage



Protective Ground  
(Earth) Terminal



ATTENTION  
Refer to Manual



Double  
Insulated

## Certifications and Compliances

**CSA Certified AC Adapter.** CSA Certification includes the AC adapters appropriate for use in the North America power network. All other AC adapters supplied are approved for the country of use.

**Compliances.** Consult the product specifications for Overvoltage Category and Safety Class.

# Preface

This is the Programmer Manual for the TDS 200-Series oscilloscopes. This manual provides information about operating your instrument using an RS-232 or GPIB interface (available with optional TDS2CM Communications Extension Module or TDS2MM Measurement Extension Module).

## Related Manuals

The following table lists related TDS 200-series oscilloscope and extension modules documentation. The service manual provides module level repair information (071-0492-XX, English).

Language	User Manual Part Number	Extension Module Instructions Part Number
English	071-0398-XX	071-0409-XX
French	071-0400-XX*	071-0483-XX
Italian	071-0401-XX*	071-0484-XX
German	071-0402-XX*	071-0485-XX
Spanish	071-0399-XX*	071-0482-XX
Japanese	071-0405-XX*	071-0488-XX
Portuguese	071-0403-XX*	071-0486-XX
Simplified Chinese	071-0406-XX*	071-0489-XX
Traditional Chinese	071-0407-XX*	071-0490-XX
Korean	071-0408-XX*	071-0491-XX
Russian	071-0404-XX	071-0487-XX

\*These manuals contain a language overlay for the front-panel controls.

## Conventions

Refer to the *Command Syntax* section of the *Syntax and Commands* chapter (page 2–1) for information about command conventions.

## Contacting Tektronix

Product support	<p>For questions about using Tektronix measurement products, call toll free in North America: 1-800-TEK-WIDE (1-800-835-9433 ext. 2400) 6:00 a.m. – 5:00 p.m. Pacific time</p> <p>Or contact us by e-mail: tm_app_supp@tek.com</p> <p>For product support outside of North America, contact your local Tektronix distributor or sales office.</p>
Service support	<p>Tektronix offers extended warranty and calibration programs as options on many products. Contact your local Tektronix distributor or sales office.</p> <p>For a listing of worldwide service centers, visit our web site.</p>
For other information	<p>In North America: 1-800-TEK-WIDE (1-800-835-9433) An operator will direct your call.</p>
To write us	<p>Tektronix, Inc. P.O. Box 1000 Wilsonville, OR 97070-1000 USA</p>
Web site	<p><a href="http://www.Tektronix.com">www.Tektronix.com</a></p>



# Getting Started





# Getting Started

Before you can use this programming manual you must have installed a TDS2CM or TDS2MM Extension Module onto your TDS 200-Series oscilloscope. Follow the instructions in the *TDS 200-Series Extension Module Instructions Manual* to install, test, and configure your extension module.

Refer to the *TDS 200-Series Digital Real-Time Oscilloscope User Manual* for general information on how to operate the oscilloscope.







# Syntax and Commands



# Command Syntax

You can control the oscilloscope through the GPIB or RS-232 interface using a large group of commands and queries. This section describes the syntax these commands and queries use and the conventions the oscilloscope uses to process them. The commands and queries themselves are listed in the *Command Descriptions* section.

You transmit commands to the oscilloscope using the enhanced American Standard Code for Information Interchange (ASCII) character encoding. *Appendix A* contains a chart of the ASCII character set.

The Backus-Naur Form (BNF) notation is used in this manual to describe commands and queries as shown in Table 2-1.

**Table 2-1: BNF notation**

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 simply called commands and queries). Commands change instrument settings or perform a specific action. Queries cause the oscilloscope to return data and information about its status.

Most commands have both a set form and a query form. The query form of the command is the same as the set form except that it ends with a question mark. 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 are set only and some are query only.

A few commands do both a set and query action. For example, the `*CAL?` command runs a self-calibration program on the oscilloscope, then returns the result of the calibration.

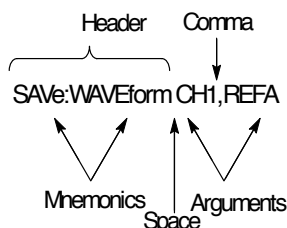
A command message is a command or query name, followed by any information the oscilloscope needs to execute the command or query. Command messages consist of five different element types, which are defined in Table 2-2 and shown in Figure 2-1.

**Table 2-2: Command message elements**

Symbol	Meaning
<Header>	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. The beginning colon can never be used with command headers beginning with a star (*).
<Mnemonic>	A header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, they are always separated from each other by a colon (:) character.
<Argument>	A quantity, quality, restriction, or limit associated with the header. Not all commands have an argument, while other commands have multiple arguments. Arguments are separated from the header by a <Space>. Arguments are separated from each other by a <Comma>.

**Table 2-2: Command message elements (Cont.)**

Symbol	Meaning
<Comma>	A single comma between arguments of multiple-argument commands. It may optionally have white space characters before and after the comma.
<Space>	A white space character between command header and argument. It may optionally consist of multiple white space characters.

**Figure 2-1: Command message elements**

## Commands

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

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

A command header is made up 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 of 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 oscilloscope to return information about its status or settings. 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, MEASUREMENT:MEAS<x>:UNITS? returns the measurement units, while MEASUREMENT:MEAS<x>:TYPE? returns the measurement type selected for the measurement, and MEASUREMENT:MEAS<x>? returns all the measurement parameters for the specified measurement.

**Headers in Query Responses**

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 instrument returns command headers as part of the query and formats the query response as a valid set command. When header is off, the instrument sends back only the values in the response. This format can make it easier to parse and extract the information from the response. Table 2–3 shows the difference in responses.

**Table 2–3: Comparison of header off and on responses**

Query	Header Off Response	Header On Response
ACQUIRE:NUMAVG?	64	:ACQUIRE:NUMAVG 64
CH1:COUPLing?	DC	:CH1:COUPLING DC

## Clearing the Output Queue

To clear the output queue and reset the oscilloscope to accept a new command or query, send a Device Clear (DCL) from a GPIB host or a break signal from an RS-232 host.

## Command Entry

Follow these general rules when entering commands:

- 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 oscilloscope ignores commands that consists of just a combination of white space characters and line feeds.

## Abbreviating Commands

You can abbreviate many oscilloscope commands. These abbreviations are shown in capital letters in the command listing in the *Command Groups* section on page 2–15 and *Command Descriptions* section on page 2–33. For example, the command ACQuire:NUMAvg can be entered simply as ACQ:NUMA or acq:numa.

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

### Concatenating Commands

You can concatenate any combination of set commands and queries using a semicolon (;). The oscilloscope executes concatenated commands in the order received. When concatenating commands and queries you must follow these rules:

- Completely different headers must be separated by both a semicolon and by the beginning colon on all commands but the first. For example, the commands `TRIGger:MODE NORMa1` and `ACQuire:NUMAVg 16` can be concatenated into a single command:

```
TRIGger:MODE NORMa1;:ACQuire:NUMAVg 16
```

- 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, the commands `ACQuire:MODE AVErAge` and `ACQuire:NUMAVg 16` could be concatenated into a single command:

```
ACQuire:MODE AVErAge; NUMAVg 16
```

The longer version works equally well:

```
ACQuire:MODE AVErAge;:ACQuire:NUMAVg 16
```

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

```
ACQuire:MODE AVErAge;*TRG
```

The instrument processes commands that follow the star command as if the star command was not there, so:

```
ACQuire:MODE AVErAge;*TRG;NUMAVg 16
```

sets the acquisition mode to average and sets acquisition averaging to 16. The `*TRG` command is ignored.

- When you concatenate queries, the responses to all queries are combined into a single response message. For example, if channel 1 coupling is set to DC and the bandwidth is set to 20MHz, the concatenated query:

```
CH1:COUPling?;BANdwidth?
```

returns `:CH1:COUPLING DC;:CH1:BANDWIDTH ON` if header is on, or `DC;ON` if header is off.



- You can concatenate set commands and queries in the same message. For example:

`ACquire:MODE AVErage;NUMAVg?;STATE?`

is a valid message that sets the acquisition mode to normal, queries the number of acquisitions for averaging, and then queries the acquisition state. The oscilloscope executes concatenated commands and queries in the order it receives them.

- Any query that returns arbitrary data, such as `ID?`, must be the last query when part of a concatenated command. If the query is not last, the oscilloscope generates event message 440.

Here are some INVALID concatenation examples:

- `CH1:COUPling DC;ACquire:NUMAVg 16`  
(missing colon before `ACquire`)
- `CH1:COUPling DC;;BANDwidth ON`  
(invalid colon before `BANDwidth`)
- `CH1:COUPling DC;:*TRG`  
(invalid colon before a star (\*) command)
- `HORizontal:MAIn:POStion 0;MAIn:SCAlE 1E-13`  
(levels of mnemonics are different—either remove the second occurrence of `MAIn:`, or put `:HORizontal:` in front of `MAIN:SCAlE`)

## Message Terminators

This manual uses the term `<EOM>` (End of message) to represent a message terminator.

**GPIB End of Message Terminators.** GPIB EOM terminators can 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 oscilloscope always terminates messages with LF and EOI. White space is allowed before the terminator; for example, CR LF is acceptable.

**RS-232 End of Message Terminators.** RS-232 EOM terminators can be a CR (carriage return), LF (line feed), CRLF (carriage return followed by a line feed), or LFCR (line feed followed by a carriage return). When receiving, the oscilloscope accepts all four combinations as valid input message terminators regardless of the currently selected terminator. When a combination of multiple characters is selected (CRLF or LFCR), the oscilloscope interprets the first character as the terminator and the second character as a null command.

## Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic could be CH2. You can use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:VOLTs command and there is also a CH2:VOLTs command. In the command descriptions, this list of choices is abbreviated CH<x>.

### Channel Mnemonics

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

Symbol	Meaning
CH<x>	TDS 210 and TDS 220: A channel specifier; <x> is 1 or 2.
	TDS 224: A channel specifier; <x> is 1, 2, 3, or 4.

### Reference Waveform Mnemonics

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

Symbol	Meaning
REF<x>	TDS 210 and TDS 220: A reference waveform specifier; <x> is A or B.
	TDS 224: A channel specifier; <x> is A, B, C, or D.

## Waveform Mnemonics

In some commands you can specify a waveform without regard to its type: channel waveform, math waveform, or reference waveform. The “y” is the same as “x” in Reference Waveform Mnemonics.

Symbol	Meaning
<wfm>	Can be CH<x>, MATH, or REF<y>

## Cursor Position Mnemonic

When the oscilloscope displays cursors, commands may specify which cursor of the pair to use.

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

## Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. The instrument can display up to four automated measurements. The displayed measurements are specified in this way:

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is 1, 2, 3, or 4.

## Argument Types

A command argument can be in one of several forms. The individual descriptions of each command tell which argument types to use with that command.

### Numeric Arguments

Many oscilloscope commands require numeric arguments. The next table lists the three types of numeric argument.

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

The syntax shown is the data format that the oscilloscope returns in response to a query. This format is also the preferred format when sending a command to the oscilloscope.

When you enter an incorrect numeric argument, the oscilloscope automatically forces the numeric argument to a correct value. The following table lists how the oscilloscope handles incorrect numeric arguments.

Argument value	Oscilloscope response
Numeric argument is less than lowest correct value for that command	Sets the specified command to the lowest correct value and executes the command
Numeric argument is greater than the highest correct value for that command	Sets the specified command to the highest correct value and executes the command
Numeric value is between two correct values	Rounds the entered value to the nearest correct value and executes the command

### Quoted String Arguments

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by single quotes (') or double quotes ("). For example:

"this is a quoted string"

Symbol	Meaning
<QString>	Quoted string of ASCII text

Follow these rules when you use quoted strings:

1. A quoted string can include any character defined in the 7-bit ASCII character set. Refer to *Appendix A*.
2. Use the same type of quote character to open and close the string:  
`"this is a valid string"`
3. You can mix quotation marks within a string as long as you follow the previous rule:  
`"this is an 'acceptable' string"`
4. You can include a quote character within a string simply by repeating the quote. For example,  
`"here is a "" mark"`
5. Strings can have upper or lower case characters.
6. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.
7. 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.
8. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some examples of invalid strings:

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

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

Block Arguments

Several oscilloscope commands use a block argument form. The following table describes each part of a block argument Figure 2–2 shows an example of a block argument.

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

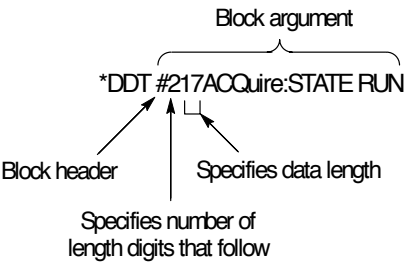


Figure 2–2: Block Argument example

<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.

#0 means that the <Block> is an indefinite length block. The <terminator> ends the block. You should not use indefinite length blocks with RS-232, because there is no way to include a <terminator> character as a <DChar> character.

The first occurrence of a <terminator> character signals the end of the block and any subsequent <DChar> characters will be interpreted as a syntax error. With the GPIB, the EOI line signals the last byte.





# Command Groups

This section lists the commands organized by functional group. The *Command Descriptions* section, starting on page 2–33, lists all commands alphabetically.

The oscilloscope GPIB and RS-232 interfaces conform to Tektronix standard codes and formats except where noted. The GPIB interface also conforms to IEEE Std 488.2-1987 except where noted.

## Acquisition Commands

Acquisition commands affect the acquisition of waveforms. These commands control mode, averaging, enveloping, and single-waveform acquisition.

**Table 2–4: Acquisition Commands**

Header	Description
ACQuire?	Return acquisition parameters
ACQuire:MODE	Set/query acquisition mode
ACQuire:NUMACq?	Return # of acquisitions obtained
ACQuire:NUMAVg	Set/query number of acquisitions for average
ACQuire:STATE	Start or stop acquisition system
ACQuire:STOPAfter	Set/query acquisition control

## Calibration and Diagnostic Commands

Calibration and Diagnostic commands let you initiate the oscilloscope self-calibration routines and examine the results of diagnostic tests.

**Table 2-5: Calibration and Diagnostic Commands**

Header	Description
*CAL?	Perform an internal self-calibration and return result status
CALibrate:ABORT	Stop an in-progress factory calibration
CALibrate:INTERNAL	Perform an internal self-calibration
CALibrate:FACTory	Initialize the factory calibration sequence
CALibrate:CONTINUE	Perform the next step in the factory calibration sequence
CALibrate:STATUS?	Return PASS or FAIL status of the last self- or factory-calibration operation
DIAG:RESULT:FLAG?	Return diagnostic tests status
DIAG:RESULT:LOG?	Return diagnostic test sequence results
ERRLOG:FIRST?	Returns first entry from error log
ERRLOG:NEXT?	Returns next entry from error log

## Cursor Commands

Cursor commands provide control over the oscilloscope cursor display and readout.

**Table 2-6: Cursor Commands**

Header	Description
CURSor?	Returns cursor settings
CURSor:FUNCTION	Set/query cursors on or off; select cursor type
CURSor:HBArs?	Return horizontal bar settings
CURSor:HBArs:DELTA?	Return vertical distance between horizontal bar cursors
CURSor:HBArs:POSITION<x>	Set/query position of a horizontal bar cursor
CURSor:HBArs:UNITs?	Query vertical scale units
CURSor:SElect:SOUrce	Select waveform
CURSor:VBArS?	Return vertical bar settings
CURSor:VBArS:DELTA?	Return horizontal distance between cursors
CURSor:VBArS:POSITION<x>	Set/query position of a vertical bar cursor
CURSor:VBArS:UNITs	Set/query vertical cursors to time or frequency

## Display Commands

Display commands let you change the graticule style, displayed contrast, and alter other display attributes.

**Table 2-7: Display Commands**

Header	Description
DISplay?	Returns display settings
DISplay:CONTRast	Set/query the LCD display contrast
DISplay:FORMat	Set/query YT or XY display
DISplay:PERStence	Set/query the accumulate time
DISplay:STYle	Set/query waveform display style

## Hard Copy Commands

The hard copy commands let you control the format of hard copy output and control the starting and stopping of hard copies.

**Table 2-8: Hard Copy Commands**

Header	Description
HARDCopy	Start or terminate hard copy
HARDCopy:FORMat	Set/query the hard copy output format
HARDCopy:LAYout	Set/query the hard copy orientation
HARDCopy:PORT	Set/query the hard copy port for output (RS232, GPIB, or Centronics)

## Horizontal Commands

Horizontal commands control the time bases of the oscilloscope. You can set the position and time per division of both the main and window time bases.

You can substitute SECdiv for SCALE in all appropriate horizontal commands. This provides program compatibility with previous Tektronix digitizing oscilloscopes.

**Table 2–9: Horizontal Commands**

Header	Description
HORizontal?	Return horizontal settings
HORizontal:DELay:POSition	Position window
HORizontal:DELay:SCALE	Set/query window time base time/division
HORizontal:DELay:SECdiv	Same as HORizontal:DELay:SCALE
HORizontal:MAIn	Set/query main time base time/division
HORizontal:MAIn:POSition	Set/query main time base trigger point
HORizontal:MAIn:SCALE	Set/query main time base time/division
HORizontal:MAIn:SECdiv	Same as HORizontal:MAIn:SCALE
HORizontal:POSition	Set/query position of waveform to display
HORizontal:RECOrdlength	Return waveform record length
HORizontal:SCALE	Same as HORizontal:MAIn:SCALE
HORizontal:SECdiv	Same as HORizontal:MAIn:SCALE
HORizontal:VIEW	Select view
MATH:FFT:HORizontal:POSition (TDS2MM Only)	Set/query
MATH:FFT:HORizontal:SCALE (TDS2MM Only)	Set/query

## Measurement Commands

Measurement commands control the automated measurement system. Up to four automated measurements can be displayed on the oscilloscope screen. In the commands, these four measurement readouts are named MEAS<x>, where <x> can be 1, 2, 3, or 4.

In addition to the four measurement readouts displayed, the measurement commands let you specify a fifth measurement, IMMed. The immediate measurement has no front-panel equivalent, and the instrument never displays immediate measurements. Because they are computed only when they are requested, immediate measurements slow the waveform update rate less than displayed measurements.

Use the VALue? query to obtain measurement results of either displayed or immediate measurements.

Several measurement commands set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement readout.

**Table 2-10: Measurement Commands**

Header	Description
MEASUrement?	Return all measurement parameters
MEASUrement:IMMed?	Return immediate measurement parameters
MEASUrement:IMMed:SOUrce	Set/query channel to take the immediate measurement from
MEASUrement:IMMed:TYPe	Set/query the immediate measurement to be taken
MEASUrement:IMMed:UNIts?	Return the immediate measurement units
MEASUrement:IMMed:VALue?	Return the immediate measurement result
MEASUrement:MEAS<x>?	Return parameters on the periodic measurement

**Table 2–10: Measurement Commands (Cont.)**

<b>Header</b>	<b>Description</b>
MEASUrement:MEAS<x>:SOUrce	Set/query channel to take the periodic measurement from
MEASUrement:MEAS<x>:TYPe	Set/query the type of periodic measurement to be taken
MEASUrement:MEAS<x>:UNIts?	Returns the units for periodic measurement
MEASUrement:MEAS<x>:VALue?	Returns periodic measurement results

## Miscellaneous Commands

Miscellaneous commands are a group of commands that do not fit into any other category.

Several commands and queries are common to all 488.2-1987 devices on the GPIB BUS and the device on the RS-232 interface. These commands and queries are defined by IEEE Std. 488.2-1987 and Tek Standard Codes and Formats 1989 and begin with an asterisk (\*) character.

**Table 2–11: Miscellaneous Commands**

<b>Header</b>	<b>Description</b>
AUTOSet	Automatic instrument setup
*DDT	Set/query group execute trigger (GET)
FACTory	Reset to factory default
HDR	Same as HEADER
HEADer	Set/query command header
ID?	Returns identification information
*IDN?	Returns identification information
LANGUAGE	Set/query language for display messages

**Table 2-11: Miscellaneous Commands (Cont.)**

Header	Description
LOCK	Lock front panel (local lockout)
*LRN?	Query device settings
REM	No action; remark only
*RST	Reset
SET?	Same as *LRN?
*TRG	Perform Group Execute Trigger (GET)
*TST?	Return self-test results
UNLOCK	Unlock front panel (local lockout)
VERBoSe	Return full command name or minimum spellings with query

## RS-232 Commands

RS-232 commands allow you to set or query the parameters that control the RS-232 port.

**Table 2-12: RS-232 Commands**

RS232?	Query RS232 parameters
RS232:BAUD	Set/query baud rate
RS232:HARDFlagging	Set/query hard flagging
RS232:PARity	Set/query parity type
RS232:S0FTFlagging	Set/query soft flagging
RS232:TRANsmit:TERMinator	Set/query end-of-line terminator



## Save and Recall Commands

Save and Recall commands allow you to store and retrieve internal waveforms and settings. When you “save a setting,” you save most of the settings of the oscilloscope. When you then “recall a setting,” the oscilloscope restores itself to the state it was in when you saved that setting.

To display a saved waveform, use the `SElect:<wfm>` command described on page 2–138.

**Table 2–13: Save and Recall Commands**

Header	Description
*RCL	Recall setting
RECALL:SETup	Recall saved instrument setting
*SAV	Save instrument setting
SAVE:SETup	Save instrument setting
SAVE:WAVEform	Save waveform

## Status and Error Commands

Status and error commands let you determine the status of the oscilloscope and control events.

Several commands and queries are common to all devices on the GPIB bus. These commands and queries are defined by IEEE Std. 488.2-1987 and Tek Standard Codes and Formats 1989, and begin with an asterisk (\*) character.

**Table 2–14: Status and Error Commands**

Header	Description
ALLEv?	Return all events
BUSY?	Return instrument busy status
*CLS	Clear status

**Table 2–14: Status and Error Commands (Cont.)**

Header	Description
DESE	Set/query device event status enable
*ESE	Set/query standard event status enable
*ESR?	Return standard event status register
EVENT?	Return event code
EVMsg?	Return event message
EVQty?	Return number of events in queue
*OPC	Set/query operation complete
*PSC	Set/query power-on status clear
*SRE	Set/query service request enable
*STB?	Read status byte
*WAI	Wait to continue

## Trigger Commands

Trigger commands control all aspects of oscilloscope triggering.

The two types of triggers are edge and video. Edge triggering is the default type. Edge triggering lets you acquire a waveform when the signal passes through a voltage level of your choosing. Video triggering adds the capability of triggering on video fields and lines.

**Table 2–15: Trigger Commands**

Header	Description
TRIGger	Force trigger event
TRIGger:MAIn	Set main trigger level to 50%; Query returns main trigger settings
TRIGger:MAIn:EDGE?	Return edge trigger settings

**Table 2–15: Trigger Commands (Cont.)**

Header	Description
TRIGger:MAIn:EDGE:COUPling	Set/query edge trigger coupling
TRIGger:MAIn:EDGE:SLOpe	Set/query edge trigger slope
TRIGger:MAIn:EDGE:SOUrce	Set/query edge trigger source
TRIGger:MAIn:HOLDOff?	Return trigger holdoff value
TRIGger:MAIn:HOLDOff:VALue	Set/query trigger holdoff value
TRIGger:MAIn:LEVe1	Set/query trigger level
TRIGger:MAIn:MODe	Set/query trigger mode
TRIGger:MAIn:TYPe	Set/query main trigger type
TRIGger:MAIn:VIDeo?	Query video trigger parameters
TRIGger:MAIn:VIDeo:POLarity	Set/query video trigger polarity
TRIGger:MAIn:VIDeo:SOUrce	Set/query video trigger source
TRIGger:MAIn:VIDeo:SYNC	Set/query video trigger sync
TRIGger:STATE?	Return trigger system status

## Vertical Commands

Vertical commands control the attributes of the channels. The SElect:<wfm> command also displays a specified waveform or removes it from the display.

**Table 2–16: Vertical Commands**

Header	Description
CH<x>?	Return vertical parameters
CH<x>:BANDwidth	Set/query channel bandwidth
CH<x>:COUPling	Set/query channel coupling

**Table 2–16: Vertical Commands (Cont.)**

Header	Description
CH<x>:INVert *	Set/query channel invert
CH<x>:POSiTion	Set/query channel position
CH<x>:PR0be	Set/query channel probe parameters
CH<x>:SCAlE	Set/query channel volts/div
CH<x>:VOLts	Same as CH<x>:SCAlE
MATH?	Query math waveform state
MATH:DEFINE	Set/query math waveform definition
MATH:FFT:VERTical :POSiTion (TDS2MM Only)	Set/query FFT waveform vertical position
MATH:FFT:VERTical :SCAlE (TDS2MM Only)	Set/query FFT waveform vertical scale
SElect:<wfm>	Set/query waveform display state

**All oscilloscope, firmware version, and module combinations except DS 210 and TDS 220 with firmware below V 2.00 and a TDS2CM.**

## Waveform Commands

Waveform commands let you transfer waveform data points to and from the oscilloscope. 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 peak-detect waveforms, each data value is either the min or max of a min/max pair. Before you can transfer waveform data, you must specify the data format and waveform locations.

Refer to the text following this table for more information about waveform commands.

**Table 2–17: Waveform Commands**

Header	Description
CURVe	Transfer waveform data to or from the instrument
DATA	Set/query the waveform data format and location
DATA:DESTination	Set/query the destination for waveforms sent to the oscilloscope
DATA:ENCdg	Set/query the waveform data encoding method
DATA:SOUrce	Set/query the source of CURVe? data
DATA:STARt	Set/query the starting point in waveform transfer
DATA:STOP	Set/query the ending point in waveform transfer
DATA:TARget	Same as DATA:DESTination
DATA:WIDth	Set/query the byte width of waveform points
WAVFrm?	Returns waveform preamble and curve data
WFMPre?	Returns waveform preamble
WFMPre:BIT_Nr	Set/query the preamble bit width of waveform points
WFMPre:BN_Fmt	Set/query the preamble binary encoding type
WFMPre:BYT_Nr	Set/query the preamble byte width of waveform points
WFMPre:BYT_Or	Set/query the preamble byte order of waveform points
WFMPre:ENCdg	Set/query the preamble encoding method

**Table 2–17: Waveform Commands (Cont.)**

Header	Description
WFMPre:NR_Pt	Query the number of points in the curve transfer
WFMPre:PT_Fmt	Set/query the format of curve points
WFMPre:PT_Off	Query the trigger offset
WFMPre:WFIId?	Query the waveform identifier
WFMPre:XINcr	Set/query the horizontal sampling interval
WFMPre:XUNit	Set/query the horizontal units
WFMPre:XZErO	Set/query time of first point in waveform
WFMPre:YMUlt	Set/query the vertical scale factor
WFMPre:YOOff	Set/query the vertical offset
WFMPre:YUNit	Set/query the vertical units
WFMPre:YZErO?	Query the 0 dB level (TDS2MM Only)
WFMPre:<wfm>?	Returns waveform formatting data
WFMPre:<wfm>:PT_Fmt	Set/query the format of curve points
WFMPre:<wfm>:PT_Off?	Query the trigger offset
WFMPre:<wfm>:WFIId?	Query the waveform identifier
WFMPre:<wfm>:XINcr	Set/query the horizontal sampling interval
WFMPre:<wfm>:XUNit	Set/query the horizontal units
WFMPre:<wfm>:XZErO	Set/query the time of first datapoint in waveform
WFMPre:<wfm>:YMUlt	Set/query the vertical scale factor
WFMPre:<wfm>:YOOff	Set/query the vertical position
WFMPre:<wfm>:YUNit	Set/query the vertical units
WFMPre:<wfm>:YZErO?	Query the 0 dB level (TDS2MM Only)

## Waveform Data Formats

Waveform data uses one 8-bit data byte to represent each data point, regardless of the acquisition mode.

For compatibility with earlier digitizing oscilloscopes, the DATA:WIDTH command lets you specify the number of bytes per data point when transferring data to and from the oscilloscope. If you specify two bytes for data that uses only one, the least significant byte will be filled with zeros. The oscilloscope ignores the least-significant byte of incoming waveforms.

The oscilloscope can transfer waveform data in either ASCII or binary format. Use the DATA:ENCdg command to specify one of the following formats:

- ASCII data is represented by signed integer values. The range of values depends on the byte width specified. One-byte-wide data ranges from -128 to 127. Two-byte-wide data ranges from -32768 to 32767.

Each data value requires two to seven characters. This includes one character for the minus sign if the value is negative, one to five ASCII characters for the waveform value, and a comma to separate data points.

An example of an ASCII waveform data string follows:

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

- Binary data can be represented by signed integer or positive integer values. The range of the values depends on the byte width specified. Table 2–18 lists the ranges for one- and two-byte-wide data.

**Table 2–18: Binary data ranges**

Byte width	Signed integer range	Positive integer range
1	-128 to 127	0 to 255
2	-32,768 to 32,767	0 to 65,535

The defined binary formats also specify the order in which the bytes are transferred giving a total of four binary formats: RIBinary, RPBinary, SRIBinary, and SRPbinary.

RIBinary is signed integer where the most significant byte is transferred first, and RPBinary is positive integer where the most significant byte is transferred first. SRIBinary and SRPbinary correspond to RIBinary and RPBinary respectively but use a swapped byte order where the least significant byte is transferred first. The byte order is ignored when DATA:WIDTH is set to 1.

**Waveform Data Record**

You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. The DATA:START and DATA:STOP commands let you specify the first and last data points of the waveform record.

When transferring data into the oscilloscope you must specify the location of the first data point within the waveform record. For example, when DATA:START is set to 1, data points will be stored starting with the first point in the record, and when DATA:START is set to 500, data will be stored starting at the 500<sup>th</sup> point in the record. The oscilloscope ignores DATA:STOP when reading in data as the oscilloscope will stop reading data when there is no more data to read or when it has reached 2500 data points.



You must specify the first and last data points in the waveform record when transferring data from the oscilloscope to an external device. Setting DATA:START to 1 and DATA:STOP to 2500 always sends the entire waveform, regardless of the acquisition mode.

### **Waveform Data Locations and Memory Allocation**

The DATA:SOURce command specifies the location of the data when transferring waveforms from the oscilloscope. You can transfer one waveform at a time.

You can transfer only one waveform into the oscilloscope at a time. Each waveform sent to the oscilloscope is stored in one of two stored waveform locations. You specify the stored waveform location with the DATA:DESTination command.

---

**NOTE** *The oscilloscope accepts waveforms that are  $\leq 2500$  data points long. The oscilloscope truncates waveforms longer than 2500 data points.*

---

### **Waveform Preamble**

Each waveform that is transferred has an associated waveform preamble that contains information such as the horizontal scale, vertical scale, and other settings in place when the waveform was created. Refer to the WFMPre commands starting on page 2–157 for more information about the waveform preamble.

### **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**

Data transfer times depend on data format data width and the speed of the controller. To transfer waveforms from the oscilloscope to an external controller, do the following steps.

1. Use the DATA:SOURce command to select the waveform source.
2. Use the DATA:ENCdg command to specify the waveform data format.
3. Use the DATA:WIDth command to specify the number of bytes per data point.
4. Use the DATA:STARt and DATA:STOP commands to specify the portion of the waveform that you want to transfer.
5. Use the WFMPRe? command to transfer waveform preamble information.
6. Use the CURVe? command to transfer waveform data.

### **Transferring Waveform Data To the Oscilloscope**

To transfer waveform data to one of two oscilloscope waveform storage locations, do the following steps:

1. Use the DATA:DESTination command to specify the stored waveform location.
2. Use the DATA:ENCdg command to specify the waveform data format.
3. Use the DATA:WIDth command to specify the number of bytes per data point.
4. Use the DATA:STARt command to specify the first data point in the waveform record.
5. Use the WFMPRe command to transfer waveform preamble information.
6. Use the CURVe command to transfer waveform data.

# Command Descriptions

Commands either set or query oscilloscope values. Some commands both set and query, some only set, and some only query. This manual marks set-only commands with the words “No Query Form.” It marks query-only commands with a question mark appended to the command and the words “Query Only” in the command name.

This manual fully spells out headers, mnemonics, and arguments with the minimal spelling shown in upper case. For example, to use the abbreviated form of the ACQUIRE:MODE command just type ACQ:MOD.

The syntax of some commands varies, depending on which extension module and oscilloscope you are using. The differences are noted.

---

**NOTE** *While trigger view is active (by pushing the TRIGGER VIEW button on the front panel), the oscilloscope ignores the set form of most commands. If you send a command at this time the oscilloscope generates execution error 221 (Settings conflict).*

---

## ACQUIRE? (Query Only)

Returns all the current acquisition parameters.

### Group

Acquisition

### Syntax

ACQUIRE?

### Returns

Returns current acquisition parameters

### Examples

ACQUIRE?

might return the string :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMAVG 16 for the current acquisition parameters.

### Related Commands

None

## ACQUIRE:MODE

Sets or queries the oscilloscope acquisition mode. This affects all live waveforms. This command is equivalent to setting Mode in the Acquire menu.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration that is determined by the horizontal scale (time per division).

The oscilloscope sampling system can operate at a rate greater than that indicated by the horizontal scale. Therefore an acquisition interval can include more than one sample.

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

### Group

Acquisition

### Syntax

ACQUIRE:MODE { SAMPLE | PEAKdetect | AVERAGE }

ACQUIRE:MODE?

### Arguments

SAMPLE specifies that the displayed data point value is the first sampled value that was taken during the acquisition interval. The waveform data has 8 bits of precision in all acquisition modes. You can request 16 bit data with a CURVE? query, but the lower-order 8 bits of data will be zero. SAMPLE is the default mode.

PEAKdetect specifies the display of the high-low range of the samples taken from a single waveform acquisition. The oscilloscope displays the high-low range as a vertical range that extends from the highest to the lowest value sampled during the acquisition interval. PEAKdetect mode can reveal the presence of aliasing.

AVERage specifies averaging mode, where the resulting waveform shows an average of SAMPLE data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the average waveform is set or queried using the ACQUIRE:NUMAVg command.

### Examples

ACQUIRE:MODE PEAKdetect  
displays a vertical area representing the range of the highest to lowest value of the acquired signal.

ACQUIRE:MODE?  
might return SAMPLE.

### Related Commands

ACQUIRE:NUMAVg, CURVe?, DATA:WIDTH

## ACQUIRE:NUMAcq? (Query Only)

Indicates the number of acquisitions that have taken place since starting oscilloscope acquisition. The maximum number of acquisitions that can be counted is  $2^{31}-1$ . This value is reset to zero when you change any Acquisition, Horizontal, Vertical, or Trigger arguments that affect the waveform, with the following exceptions:

- Changing the vertical position, trigger level, or trigger holdoff when in Sample or Peak Detect mode does not reset the value to zero.
- Changing the vertical scale during Scan does not reset the value to zero.

Note that any change made when in Average mode aborts the acquisition and resets ACQUIRE:NUMACq to zero.

### **Group**

Acquisition

### **Syntax**

ACQUIRE:NUMACq?

### **Returns**

<NR1>

### **Examples**

ACQUIRE:NUMACQ?

might return 350, indicating that 350 acquisitions took place since an ACQUIRE:STATE RUN command was executed.

### **Related Commands**

ACQUIRE:STATE

## **ACQUIRE:NUMAVg**

Sets the number of oscilloscope waveform acquisitions that make up an averaged waveform. This command is equivalent to setting the **Average** count in the Acquire menu.

### **Group**

Acquisition

### **Syntax**

ACQUIRE:NUMAVg <NR1>

ACQUIRE:NUMAVg?

### **Arguments**

<NR1> is the number of waveform acquisitions. Correct values are 4, 16, 64, and 128.

**Examples**

ACQUIRE:NUMAVG 16

specifies that an averaged waveform will show the result of combining 16 separately acquired waveforms.

ACQUIRE:NUMAVG?

might return 64, indicating that there are 64 acquisitions specified for averaging.

**Related Commands**

ACquire:MODE

**ACquire:STATE**

Starts or stops oscilloscope acquisitions. This command is the equivalent of pressing the front-panel **RUN/STOP** button. If ACquire:STOPAfter is set to SEQUENCE, other signal events may also stop acquisition.

---

**NOTE** The best way to determine when a single sequence acquisition is complete is to use \*OPC? instead of ACquire:STATE?.

---

**Group**

Acquisition

**Syntax**

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

ACquire:STATE?

**Arguments**

OFF or STOP or <NR1> = 0 stops acquisitions.

ON or RUN or <NR1>  $\neq$  0 starts acquisition and display of waveforms. If the command was issued in the middle of an acquisition sequence (for instance averaging or enveloping), RUN restarts the sequence, discarding any data accumulated before the STOP. It also resets the number of acquisitions.

### Examples

**ACQUIRE:STATE RUN**

starts acquisition of waveform data and resets the number of acquisitions count (NUMACq) to zero.

**ACQUIRE:STATE?**

returns either 0 or 1, depending on whether the acquisition system is running.

### Related Commands

ACquire:NUMACq?, ACquire:STOPAfter, \*OPC?

## ACquire:STOPAfter

Tells the oscilloscope when to stop taking acquisitions for edge trigger type.

### Group

Acquisition

### Syntax

ACquire:STOPAfter { RUNSTop | SEquence}

ACquire:STOPAfter?

### Arguments

RUNSTop specifies that the run and stop state should be determined by the user pressing the front-panel RUN/STOP button.

SEquence specifies “single sequence” operation, where the oscilloscope stops after it has acquired enough waveforms to satisfy the conditions of the acquisition mode. For example, if the acquisition mode is set to sample, the oscilloscope stops after digitizing a waveform from a single trigger event. However, if the acquisition mode is set to average 64 waveforms, then the oscilloscope stops only after acquiring all 64 waveforms. The ACquire:STATE command and the front-panel **RUN/STOP** button also stop acquisitions when the oscilloscope is in single sequence mode.



**Examples**

ACQUIRE:STOPAFTER RUNStop  
sets the oscilloscope to stop acquisition when the user presses the front-panel **RUN/STOP** button.

ACQUIRE:STOPAFTER?  
might return SEQUENCE.

**Related Commands**

ACquire:MODE, ACquire:STATE

## ALLEv? (Query Only)

Causes the oscilloscope to return all events and their messages, and removes the returned events from the Event Queue. The messages are separated by commas. Use the \*ESR? query to enable the events to be returned. For a complete discussion of the use of these registers, see page 3–1. This command is similar to repeatedly sending \*EVMsg? queries to the instrument.

**Group**

Status and error

**Syntax**

ALLEv?

**Returns**

The event code and message in the following format:

```
<Event Code><Comma><QString>[<Comma><Event  
Code><Comma><QString>...]
```

```
<QString>::= <Message>;[<Command>]
```

<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 is returned as possible without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

### Examples

ALLEV?

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

### Related Commands

\*CLS, DESE, \*ESE, \*ESR?, EVENT?, EVMsg?, EVQty?, \*SRE, \*STB?

## AUTOSet (No Query Form)

Causes the oscilloscope to adjust its vertical, horizontal, and trigger controls to display a stable waveform. This command is equivalent to pushing the front-panel **AUTOSET** button.

For a detailed description of the autoset function, refer to the *TDS 210 & TDS 220 User Manual*.

### Group

Miscellaneous

### Syntax

AUTOSet { EXECUTE }

### Arguments

EXECute autosets the displayed waveforms.

## BUSY? (Query Only)

Returns the status of the oscilloscope. This command allows you to synchronize the operation of the oscilloscope with your application program. Synchronization methods are described on page 3–10.

### Group

Status and error

### Syntax

BUSY?

**Returns**

A number 0 means that the oscilloscope is not busy processing any of the commands listed in Table 2–21 (\*OPC) on page 2–124.

A number 1 means that the oscilloscope is busy processing one of the commands listed in Table 2–21 (\*OPC) on page 2–124.

**Examples**

BUSY?

might return 1, indicating that the oscilloscope is busy.

**Related Commands**

\*OPC, \*WAI

## **\*CAL? (Query Only)**

Performs an internal self-calibration and returns its status. This is equivalent to selecting Do Self Cal in the Utility menu. Although \*CAL? is a query command, it does perform an action.

---

**NOTE** *The self-calibration can take several minutes to complete. The oscilloscope does not execute any commands until the self-calibration is complete.*

---

### **Group**

Calibration and Diagnostic

### **Syntax**

\*CAL?

### **Returns**

0 indicates that the self-calibration completed without any errors detected.

Any value other than zero indicates that the self-calibration did not complete successfully or completed with errors.

### **Examples**

\*CAL?

performs a self-calibration and might return 0 to indicate that it completed successfully.

### **Related Commands**

CALibrate:INTERNAL, CALibrate:STATUS?

## **CALibrate:ABORT (No Query Form)**

---

**NOTE** *This command should only be used in a qualified service environment. For more information about the factory calibration sequence, refer to the TDS 200-Series Digital Real-Time Oscilloscope Service Manual.*

---

Aborts the factory calibration process. When you abort the factory calibration, the oscilloscope restores the calibration settings to the previous factory calibration constants stored in non-volatile memory.

### **Group**

Calibration and Diagnostic

### **Syntax**

CALibrate:ABORT

### **Examples**

CALibrate:ABORT

stops the in-process factory calibration procedure.

### **Related Commands**

CALibrate:CONTINUE, CALibrate:FACTory, CALibrate:INTERNAL, CALibrate:STATUS?

## **CALibrate:CONTINUE (No Query Form)**

---

**NOTE** *This command should only be used in a qualified service environment. For more information about the factory calibration sequence, refer to the TDS 200-Series Digital Real-Time Oscilloscope Service Manual.*

---

Performs the next step in the factory calibration operation.

**Group**

Calibration and Diagnostic

**Syntax**

CALibrate:CONTINUE

**Examples**

CALibrate:CONTINUE

performs the next step in the factory calibration operation.

**Related Commands**

\*CAL?, CALibrate:ABORT, CALibrate:FACTory,  
CALibrate:INTERNAL, CALibrate:STATUS?

## CALibrate:FACTory (No Query Form)

---

**NOTE** *This command should only be used in a qualified service environment. For more information about the factory calibration sequence, refer to the TDS 200-Series Digital Real-Time Oscilloscope Service Manual.*

---

Starts the oscilloscope's internal factory calibration operation. The calibration operation consists of a sequence of steps. You send the CALibrate:CONTINUE command to advance to the next calibration step. The calibration program automatically sets up the oscilloscope for each calibration step. Use the CALibrate:ABORT command to abort the factory calibration.

You can only send synchronization commands or queries (such as \*OPC, OPC?, \*WAI, BUSY?) while doing a factory calibration.

**Group**

Calibration and Diagnostic

**Syntax**

CALibrate:FACTory

**Examples**

CALibrate:FACTory

starts the factory calibration process.

**Related Commands**

\*CAL?, CALibrate:ABORT, CALibrate:CONTINUE, CALibrate:INTERNAL, CALibrate:STATUS?

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

Performs an internal self-calibration but does not return any status. This is equivalent to selecting Do Self Cal in the Utility menu.

---

**NOTE** *The self-calibration can take several minutes to complete. The oscilloscope does not execute any commands until the self-calibration is complete.*

---

**Group**

Calibration and Diagnostic

**Syntax**

CALibrate:INTERNAL

**Examples**

CALibrate:INTERNAL

performs an internal self-calibration.

**Related Commands**

\*CAL?, CALibrate:ABORT, CALibrate:CONTINUE, CALibrate:FACTory, CALibrate:STATUS?

## **CALibrate:STATUS? (Query Only)**

Returns the status of the last calibration operation performed (either self- or factory-calibration) since power up.

### **Group**

Calibration and Diagnostic

### **Syntax**

CALibrate:STATUS?

### **Returns**

PASS indicates that the oscilloscope completed the last calibration operation without detecting any errors.

FAIL indicates that the oscilloscope detected errors during the last calibration operation, or that no calibration operations have been performed since power up.

### **Examples**

CALibrate:STATUS?

might return CALIBRATE:STATUS FAIL if the oscilloscope failed the last calibration operation.

### **Related Commands**

\*CAL?, CALibrate:ABORT, CALibrate:CONTINUE,  
CALibrate:FACTory, CALibrate:INTERNAL

## **CH<x>? (Query Only)**

Returns the oscilloscope vertical parameters. Because CH<x>:SCALE and CH<x>:VOLts are identical, only CH<x>:SCALE is returned.

### **Group**

Vertical

### **Syntax**

CH<x>?



**Returns**

Oscilloscope vertical parameters

**Examples**

CH1?

might return the string :CH1:SCALE 1.0E0;POSITION 0.0E0;  
COUPLING DC;BANDWIDTH OFF;PROBE 1.0E0  
for channel 1.

**Related Commands**

None

## **CH<x>:BANDwidth**

Sets or queries the bandwidth setting of the specified oscilloscope channel. This command is equivalent to setting BW Limit in the Vertical menu.

**Group**

Vertical

**Syntax**

CH<x>:BANDwidth { ON | OFF }

CH<x>:BANDwidth?.

**Arguments**

ON sets the channel bandwidth to 20 MHz.

OFF sets the channel bandwidth to the full bandwidth of the oscilloscope. In most acquisition modes, full bandwidth is 60MHz or 100MHz (depending on the oscilloscope model). There are two exceptions. At vertical scales of 5mv/div or less (sensitivity at the BNC; that is, after the probe factor is removed), the full bandwidth is 20MHz. When the acquisition mode is Peak Detect, and the vertical scale at the BNC is 10mv/div or less, the full bandwidth is also 20MHz.

### Examples

CH2:BANDWIDTH ON

sets the bandwidth of channel 2 to 20 MHz.

CH1:BANDWIDTH?

might return OFF, which indicates that there is no bandwidth limiting on channel 1.

### Related Commands

None

## CH<x>:COUPLing

Sets or queries the input attenuator coupling setting of the specified oscilloscope channel. This command is equivalent to setting Coupling in the Vertical menu.

### Group

Vertical

### Syntax

CH<x>:COUPLing { AC | DC | GND }

CH<x>:COUPLing?

### Arguments

AC sets the specified oscilloscope channel to AC coupling.

DC sets the specified oscilloscope channel to DC coupling.

GND sets the specified oscilloscope channel to ground. Only a flat ground-level waveform is displayed.

### Examples

CH1:COUPLING AC

establishes AC coupling on channel 1.

CH2:COUPLING?

might return DC, indicating that channel 2 is set to DC coupling.

## **CH<x>:INVert (All Oscilloscope, Firmware Version, and Module Combinations Except TDS 210 or TDS 220 with Firmware Below V 2.00 and a TDS2CM)**

Sets or queries the inversion state of the specified oscilloscope channel. This command is equivalent to setting Invert in the Vertical channel menus.

### **Group**

Vertical

### **Syntax**

CH<x>:INVert { ON | OFF }

CH<x>:INVert?

### **Arguments**

ON inverts the specified oscilloscope channel.

OFF sets the specified oscilloscope channel to non-inverted.

### **Examples**

CH1:INVERT ON  
inverts the signal on channel 1.

CH2:INVERT?  
might return OFF, indicating that channel 2 is not inverted.

### **Related Commands**

None

## **CH<x>:POSition**

Sets or queries the vertical position of the specified oscilloscope channel. The position voltage value is applied to the signal before

digitization. This command is equivalent to adjusting the front-panel VERTICAL POSITION knob.

**Group**

Vertical

**Syntax**

CH<x>:POSition <NR3>

CH<x>:POSition?

**Arguments**

<NR3> is the position in divisions from the center graticule. The range of positions is listed in Table 2–19.

**Table 2–19: Vertical position ranges using a 1X probe**

CH<x>:SCALE	Position range
2 mV/div	±1000 divs
5 mV/div	±400 divs
10 mV/div	±400 divs
20 mV/div	±400 divs
50 mV/div	±400 divs
100 mV/div	±400 divs
200 mV/div	±10 divs
500 mV/div	±100 divs
1 V/div	±100 divs
2 V/div	±100 divs
5 V/div	±10 divs

**Examples**

CH2:POSITION 1.32E0

positions the channel 2 input signal 1.32 divisions above the center of the display.

CH1:POSITION?

might return -1.32E0, indicating that the current position of channel 1 is at -1.32 divisions.

**Related Commands**

None

**CH<x>:PRObe**

Sets or queries the attenuation factor of the specified channel.

**Group**

Vertical

**Syntax**

CH<x>:PRObe { 1 | 10 | 100 | 1000 }

CH<x>:PRObe?

**Arguments**

1 sets the specified oscilloscope channel to x1 attenuation.

10 sets the specified oscilloscope channel to x10 attenuation.

100 sets the specified oscilloscope channel to x100 attenuation.

1000 sets the specified oscilloscope channel to x1000 attenuation.

**Returns**

<NR1>

**Examples**

CH2:PROBE 1000

sets channel 2 to x1000 attenuation.

CH1:PRObe?

might return CH1:PROBE 10.

**Related Commands**

None

## **CH<x>:SCALE**

Sets or queries the vertical gain of the specified oscilloscope channel. This command is equivalent to adjusting the front-panel VOLTS/DIV knob.

**Group**

Vertical

**Syntax**

CH<x>:SCALE <NR3>

CH<x>:SCALE?

**Arguments**

<NR3> is the gain, in volts per division. The range is 5 V/div to 2 mV/div when using a 1X probe.

**Examples**

CH1:SCALE 100E-3  
sets the channel 1 gain to 100 mV/div.

CH2:SCALE?  
might return 1.0E0, indicating that the current V/div setting of channel 2 is 1 V/div.

**Related Commands**

CH1:VOLts

## **CH<x>:VOLts**

Sets or queries the vertical gain of the specified channel. This command is identical to the CH<x>:SCALE command and is included for compatibility purposes. Only CH<x>:SCALE is returned in response to a CH<x>? query.

### **Group**

Vertical

### **Syntax**

CH<x>:VOLts <NR3>

CH<x>:VOLts?

### **Arguments**

<NR3> is the gain, in volts per division. The range is 5 V/div to 2 mV/div when using a 1X probe.

### **Examples**

CH1:VOLTS 100E-3

sets the channel 1 gain to 100 mV/div.

CH2:VOLTS?

might return 1.0E0, indicating that the current V/div setting of channel 2 is 1 V/div.

### **Related Commands**

CH1:SCALE

## **\*CLS (No Query Form)**

Clears the oscilloscope status data structures.

### **Group**

Status and Error

### **Syntax**

\*CLS

The \*CLS command clears the following:

- the Event Queue
- the Standard Event Status Register (SESR)
- the Status Byte Register (except the MAV bit; see below)

If the \*CLS command immediately follows a <E0I>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates information is in the output queue. The device clear (DCL) GPIB control message will clear the output queue and also MAV.

\*CLS does not clear the output queue or MAV. \*CLS can suppress a service request that is to be generated by an \*OPC command. This will happen if a hardcopy output or single sequence acquisition operation is still being processed when the \*CLS command is executed.

A complete discussion of these registers and bits, and of event handling in general, begins on page 3–1.

### **Arguments**

None

### **Examples**

None

### **Related Commands**

DESE, \*ESE, \*ESR?, EVENT?, EVMsg?, \*SRE, \*STB?



## **CURSor? (Query Only)**

Returns all current oscilloscope cursor settings.

### **Group**

Cursor

### **Syntax**

CURSor?

### **Returns**

Oscilloscope cursor parameters

### **Examples**

CURSOR?

might return CURSOR:FUNCTION VBARS;SELECT:SOURCE CH1;  
:CURSOR:VBARS:UNITS SECONDS;POSITION1 0.0E0;POSITION2  
4.48E-3;:CURSOR:HBARS:UNITS VOLTS;POSITION1 0.0E0;POSI-  
TION2 5.0E-1 as the current cursor settings.

### **Related Commands**

None

## **CURSor:FUNCTION**

Selects and displays the oscilloscope cursor type. Cursors are attached to the waveform selected by CURSor:SElect:SOUrce. This command is equivalent to setting Type in the Cursor menu. Setting the function to anything other than OFF causes the cursor menu to be displayed.

---

**NOTE** Setting the display format to XY turns cursors off. Sending the CURSor:FUNCTION command when the display format is XY causes the oscilloscope to generate event 221 (settings conflict) and leaves the display in XY format.

---

**Group**

Cursor

**Syntax**

CURSor:FUNCTION { HBArS | OFF | VBArS }

CURSor:FUNCTION?

**Arguments**

HBArS specifies horizontal bar cursors that measure the vertical units in volts, divisions, or decibels (decibels are TDS2MM Only).

OFF removes the cursors from the display.

VBArS specifies vertical bar cursors that measure time or frequency.

**Examples**

CURSOR:FUNCTION VBARS

selects vertical bar type cursors.

**Related Commands**

CURSor:SElect

## **CURSor:HBArS? (Query Only)**

Returns the current settings for the oscilloscope horizontal bar cursors.

**Group**

Cursor

**Syntax**

CURSor:HBArS?

**Returns**

Current horizontal bar cursor settings.

**Examples**

CURSOR:HBARS?

might return CURSOR:HBARS:UNITS VOLTS;POSITION1 0.0E0;  
POSITION2 0.0E0.

**CURSor:HBARs:DELTA? (Query Only)**

Returns the difference (in vertical units) between the two horizontal bar cursors in the oscilloscope display.

---

**NOTE** *If the trigger view is active, this query returns 9.9E37 and generates event 221 (settings conflict).*

---

**Group**

Cursor

**Syntax**

CURSor:HBARs:DELTA?

**Returns**

<NR3>

**Examples**

CURSOR:HBARS:DELTA?

might return 5.08E0 for the difference between the two cursors.

## **CURSor:HBARs:POSITION**

Positions a horizontal bar cursor.

---

**NOTE** *If the trigger view is active, the query form returns 9.9E37 and generates event 221 (settings conflict).*

---

### **Group**

Cursor

### **Syntax**

CURSor:HBARs:POSITION<x> <NR3>

CURSor:HBARs:POSITION<x>?

### **Arguments**

<x> specifies which cursor position, and has the value of 1 or 2.

<NR3> specifies the horizontal bar cursor position, relative to ground (in volts when the units are volts), relative to the center of the screen (in divs when units are divisions), or relative to 1 V RMS (in decibels when the source is an FFT math waveform), for the waveform specified by the CURSor:SElect:SOUrce command.

The cursor position is limited to the graticule whenever an attempt is made to move it outside the graticule.

---

**NOTE** *The source always determines the measurement units.*

---

### **Examples**

CURSOR:HBARS:POSITION1 25.0E-3

positions one of the horizontal cursors at 25.0 mV (assuming the vertical units are volts).

CURSOR:HBARS:POSITION2?

might return -6.40E-2, indicating that the second horizontal bar cursor is at -64.0 mV (assuming the vertical units are volts).

## **CURSor:HBArS:UNIts? (Query Only)**

Returns the vertical scale units for the selected cursor source waveform.

### **Group**

Cursor

### **Syntax**

CURSor:HBArS:UNIts?

### **Returns**

VOLts indicates volts from ground as the unit of measure.

DIVs indicates divisions as the unit of measure, with center of screen as 0 divisions and bottom of screen as -4 divisions.

DECIBELS indicates decibels as the unit of measure, relative to a 1 V RMS sine wave.

UNKNOWN indicates the trigger view is active. This also generates event message 221, "Settings conflict".

### **Examples**

CURSOR:HBARS:UNIts?

might return CURSOR:HBARS:UNITS VOLTS.

## **CURSor:SElect:SOUrce**

Sets or queries the waveform that is the source of the vertical and horizontal scale factors used in determining cursor values. This command is equivalent to setting Source in the Cursor menu.

### **Group**

Cursor

### **Syntax**

CURSor:SElect:SOUrce <wfm>

CURSor:SElect:SOUrce?

### Arguments

<wfm> specifies the waveform data source on which cursor measurements will be taken.

### Examples

CURSOR:SElect:SOUrce CH1  
selects channel 1.

CURSOR:SElect:SOUrce?  
might return MATH.

## CURSor:VBArS? (Query Only)

Returns the current vertical bar cursor horizontal position and units settings.

### Group

Cursor

### Syntax

CURSor:VBArS?

### Examples

CURSOR:VBARS?  
might return CURSOR:VBARS:UNITS SECONDS;  
POSITION1 1.00E-6;POSITION2 9.00E-6

## **CURSor:VBArS:DELTA? (Query Only)**

Returns the time or frequency difference between the two vertical bar cursors. The units (seconds or Hertz) are specified by the CURSor:VBArS:UNIts command. If the cursor source is an FFT math waveform, CURSor:VBArS:DELTA is always in Hertz, regardless of the value set by CURSor:VBArS:UNIts.

---

**NOTE** *If the trigger view is active, this query returns 9.9E37 and generates event 221 (settings conflict).*

---

### **Group**

Cursor

### **Syntax**

CURSor:VBArS:DELTA?

### **Returns**

<NR3>

### **Examples**

CURSOR:VBARS:DELTA?

might return 8.92E-1, indicating that the time difference between the vertical bar cursors is 0.892 seconds.

### **Related Commands**

CURSor:VBArS:UNIts

## **CURSor:VBArS:POSITION**

Positions a vertical bar cursor. The unit is specified by the CURSor:VBArS:UNIts command, and can be in units of seconds or frequency (Hertz). If the cursor source is an FFT math waveform, CURSor:VBArS:POSITION is always in Hertz, regardless of the value set by CURSor:VBArS:UNIts.

---

**NOTE** *If the trigger view is active, the query form returns 9.9E37 and generates event 221 (settings conflict).*

---

### **Group**

Cursor

### **Syntax**

CURSor:VBArS:POSITION<x> <NR3>

CURSor:VBArS:POSITION<x>?

### **Arguments**

<x> specifies which cursor to position. Correct values are 1 and 2.

<NR3> specifies the cursor position in the units specified by the CURSor:VBArS:UNIts command. The position is relative to the trigger except when the cursor source is a math FFT waveform. The cursor position is limited to the graticule whenever an attempt is made to move it outside the graticule.

### **Examples**

CURSOR:VBARS:POSITION2 9.00E-6  
positions the second vertical bar cursor at 9  $\mu$ s.

CURSOR:VBARS:POSITION1?  
might return 1.00E-6, indicating the first vertical bar cursor is at 1  $\mu$ s.

### **Related Commands**

CURSor:VBArS:UNIts



## **CURSor:VBArS:UNIts**

Sets or queries the units for the vertical bar cursors.

### **Group**

Cursor

### **Syntax**

CURSor:VBArS:UNIts { SECOnds | HERTz }

CURSor:VBArS:UNIts?

### **Arguments**

SECOnds specifies units of time.

HERTz specifies units of frequency (reciprocal of time).

### **Examples**

CURSOR:VBARS:UNITS SECONDS

sets the units for the vertical bar cursors to seconds.

CURSOR:VBARS:UNITS?

returns HERTZ when the vertical bar cursor units are Hertz.

### **Related Commands**

CURSor:VBArS:DELTA?, CURSor:VBArS:POSITION<x>

## CURVe

Transfers oscilloscope waveform data to and from the oscilloscope in binary or ASCII format. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale. Refer to the WFMPre command starting on page 2–157 for information about the waveform preamble. The data format is specified by the DATA:ENCdg and DATA:WIDTH commands.

The CURVe? query sends data from the oscilloscope to an external device. The data source is specified by the DATA:SOURce command. The first and last data points that are transferred are specified by the DATA:STARt and DATA:STOP commands.

---

**NOTE** *In Scan Mode (Sec/div  $\geq$  100ms and AUTO Mode), approximately one division's worth of data points will be invalid due to the blanked moving cursor.*

---

The CURVe command sends waveform data from an external device to the oscilloscope. The data is stored in the stored waveform 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 waveform is displayed.

A description of the waveform transfer process starts on page 2–26, *Waveform Commands*.

### Group

Waveform

### Syntax

CURVe { <Block> | <asc curve> }

CURVe?

**Arguments**

<Block> is the waveform data in binary format. The waveform is formatted as: #<x><yyy><data> where <x> is the number of characters in <yyy>. For example, if <yyy> = 500, then <x> = 3, where <yyy> is the number of bytes to transfer. See *Block Arguments* on page 2–12.

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. Use the DATA:WIDTH command to set the width. <data> is the curve data.

<asc curve> is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>...] where each <NR1> represents a data point.

**Examples**

CURVE?

might return the ASCII data

```
:CURVE 13,6,3,2,-1,-9,-14,-19,-29,-35,-67,-71,-78,-62,-50,-31,-27,-7,0,12,29,39,45,43,41,47,41,38,33,26
```

**Related Commands**

DATA, WFMPRe

## DATA

Sets or queries the format and location of the waveform data that is transferred with the CURVE command. Since DATA:DESTination and DATA:TARget are equivalent, only DATA:DESTination is returned by the DATA? query.

**Group**

Waveform

**Syntax**

```
DATA { INIT }
```

DATA?

### Arguments

INIT initializes the waveform data parameters to their factory defaults.

### Examples

DATA INIT

initializes the waveform data parameters to their factory defaults:

DESTINATION = REFA (=TARGET)

ENCDG = RIBINARY

SOURCE = CH1

START = 1

STOP = 2500

WIDTH = 1

DATA?

might return the string :DATA:ENCDG RPBINARY;DESTINATION REFA; SOURCE REFB;START 1;STOP 500;WIDTH 2

### Related Commands

CURVE, WAVFrm

## DATA:DESTination

Sets or queries the reference memory location for storing oscilloscope waveform data that is transferred into the oscilloscope by the CURVE command. This command is identical to the DATA:TARGET command.

### Group

Waveform

### Syntax

DATA:DESTination REF<x>

DATA:DESTination?

### Arguments

REF<x> is the reference memory location where the waveform will be stored.

**Examples**

DATA:DESTINATION REFA

stores incoming waveform data into reference memory REFA.

DATA:DESTINATION?

might return REFB as the waveform location that is currently selected.

**Related Commands**

CURVe?, DATA:TArget

**DATA:ENCdg**

Sets or queries the format of the waveform data. This command is equivalent to setting WFMPre:ENCdg, WFMPre:BN\_Fmt, and WFMPre:BYT\_Or as shown in Table 2–20. Setting the DATA:ENCdg value causes the corresponding WFMPre values to be updated and vice versa.

**Group**

Waveform

**Syntax**

DATA:ENCdg { ASCIi | RIBinary | RPBinary | SRIBinary | SRPbinary }

DATA:ENCdg?

**Arguments**

ASCIi specifies the ASCII representation of signed integer (RIBinary) data. If this is the value at power-on, the WFMPre values for BN\_Fmt, BYT\_Or, and ENCdg are set as RP, MSB, and ASC respectively.

RIBinary specifies signed integer data-point representation with the most significant byte transferred first. This format results in the fastest data transfer rate when DATA:WIDTH is set to 2.

The range is -128 to 127 when DATA:WIDTH is 1. Zero is center screen. The range is -32768 to 32767 when DATA:WIDTH is 2. 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.

RPBinary specifies positive integer data-point representation with the most significant byte transferred first.

The range is 0 to 255 when DATA:WIDTH is 1. Center screen is 127. The range is 0 to 65,535 when DATA:WIDTH is 2. 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 is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

SRPBinary is the same as RPBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

**Table 2–20: DATA and WFMPre Parameter Settings**

DATA:ENCdg Setting	WFMPre Settings		
	:ENCdg	:BN_Fmt	:BYT_Or
ASCIi	ASC	N/A	N/A
RIBinary	BIN	RI	MSB
RPBinary	BIN	RP	MSB
SRIBinary	BIN	RI	LSB
SRPBinary	BIN	RP	LSB

**Examples**

DATA:ENCDG RPBINARY

sets the data encoding format to be positive integer where the most significant byte is transferred first.

DATA:ENCDG?

might return SRPBINARY for the format of the waveform data.

**Related Commands**

WFMPre:ENCDg, WFMPre:BN\_Fmt, WFMPre:BYT\_Or

## DATA:SOURce

Sets or queries which waveform will be transferred from the instrument by the CURVe?, WFMPre?, or WAVFrm? queries. You can transfer only one waveform at a time.

**Group**

Waveform

**Syntax**

DATA:SOURce <wfm>

DATA:SOURce?

**Arguments**

<wfm> is the location of the waveform data that will be transferred from the oscilloscope to the external device. Allowable values are CH<x>, MATH, and REF<x>.

**Examples**

DATA:SOURCE REFB

specifies that reference waveform REFB will be transferred in the next CURVE? query.

DATA:SOURCE?

might return REFA, indicating the source for the waveform data that is transferred using CURVE?.

**Related Commands**

CURVe?, WFMPre?, WAVFrm?

## **DATA:START**

Sets or queries the starting data point for waveform data transfers. This command lets you transfer partial waveforms to and from the oscilloscope.

### **Group**

Waveform

### **Syntax**

DATA:START <NR1>

DATA:START?

### **Arguments**

<NR1> is an integer value that ranges from 1 to 2500, and specifies the first data point that will be transferred. Data is transferred from <NR1> to DATA:STOP or 2500, whichever is less. When DATA:STOP is less than DATA:START, the values are swapped internally for CURVe?.

### **Examples**

DATA:START 10

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

DATA:START?

might return 214 as the first waveform data point that will be transferred.

### **Related Commands**

CURVe?, DATA:STOP



## DATA:STOP

Sets or queries the last data point in the waveform that will be transferred when executing the CURVe? command. This lets you transfer partial waveforms from the oscilloscope.

When using the CURVe command, the oscilloscope stops reading data when there is no more data to read or when the 2500 data point limit is reached.

### Group

Waveform

### Syntax

DATA:STOP <NR1>

DATA:STOP?

### Arguments

<NR1> is an integer value that ranges from 1 to 2500, and specifies the last data point that will be transferred. When DATA:STOP is less than DATA:START, the values are swapped internally for CURVe?.

If you always want to transfer complete waveforms, set DATA:START to 1 and DATA:STOP to 2500.

### Examples

DATA:STOP 150

specifies that the waveform transfer will stop at data point 150.

DATA:STOP?

might return 285 as the last data point that will be transferred.

### Related Commands

CURVe?, DATA START

## **DATA:TAReget**

Sets or queries the location for storing waveform data transferred from an external device to the oscilloscope when executing the CURVe command. This command is equivalent to the DATA:DESTination command and is included here for compatibility with older Tektronix instruments.

### **Group**

Waveform

### **Syntax**

DATA:TAReget REF<x>

DATA:TAReget?

### **Arguments**

REF<x> specifies the waveform storage location.

### **Examples**

DATA:TAReget REFA

stores incoming waveform data into the Ref A storage area

DATA:TAReget?

might return REFB as the waveform location that is currently selected.

### **Arguments**

REF<X>

### **Examples**

None

### **Related Commands**

CURVe, DATA:DESTination

## DATA:WIDTH

Sets the number of bytes per waveform data point to be transferred when executing the CURVe command. (Changing DATA:WIDTH may change the following WFMPre parameters: BIT\_Nr, BYT\_Nr, YMULt, YOFF, and YZErO.)

### Group

Waveform

### Syntax

DATA:WIDTH <NR1>

DATA:WIDTH?

### Arguments

<NR1> = 1 sets the number of bytes per waveform data point to 1 byte (8 bits).

<NR1> = 2 sets the number of bytes per waveform data point to 2 bytes (16 bits). If DATA:WIDTH is set to 2, the least significant byte is always zero.

### Examples

DATA:WIDTH 1

sets the data width to 1 byte per data point for CURVe data.

### Related Commands

CURVe, WFMPre:BIT\_Nr, WFMPre:BYT\_Nr

## \*DDT

Lets you specify a command or a list of commands to execute when the instrument receives a \*TRG command or the GET GPIB interface message. This is a special alias that \*TRG uses.

### Group

Miscellaneous

### **Syntax**

\*DDT { <Block> | <QString> }

\*DDT?

### **Arguments**

<Block> or <QString> is a complete sequence of program messages. The messages must contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands (see page 2–6). The sequence must be ≤80 characters. <Block> format is always returned as a query response.

### **Examples**

\*DDT #217ACQUIRE:STATE RUN<EOI>

specifies that the acquisition system will be started each time a

\*TRG command is sent.

### **Related Commands**

\*TRG

## **DESE**

Sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether or not 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 page 3–1.

### **Group**

Status and Error

### **Syntax**

DESE <NR1>

DESE?

**Arguments**

<NR1> is an integer value in the range from 0 to 255. The binary bits of DESER are set according to this value. 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, and so on).

The power-on default for DESER is all bits set to 1 if \*PSC is 1. If \*PSC is 0, the DESER maintains its value through a power cycle.

---

**NOTE** Setting 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 ESER. A discussion of event handling begins on page 3–1.

---

**Examples**

DESE 209

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

DESE?

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

**Related Commands**

\*CLS, \*ESE, \*ESR?, EVENT?, EVMsg?, \*SRE, \*STB?

**DIAG:RESULT:FLAG? (Query Only)**

Returns the pass/fail status from the last diagnostic test sequence execution. Use the DIAG:RESULT:LOG? query to determine which test(s) has failed.

**Group**

Calibration and Diagnostic

**Syntax**

DIAG:RESULT:FLAG?

**Returns**

PASS means that the oscilloscope passes all diagnostic tests.

FAIL means that the oscilloscope has failed at least one of the diagnostic tests.

**Examples**

DIAG:RESULT:FLAG?

returns either PASS or FAIL.

**Related Commands**

DIAG:RESULT:LOG?

## DIAG:RESULT:LOG? (Query Only)

Returns the internal results log from the last diagnostic test sequence execution. The list contains all modules and module interfaces that were tested along with the pass/fail status of each.

**Group**

Calibration and Diagnostic

**Syntax**

DIAG:RESULT:LOG?

**Returns**

<QString> in the following format:

<Status>,<Module name>[,<Status>,<Module name>...]

**Examples**

DIAG:RESULT:LOG?

might return "pass--CPU, pass--ACQ1, pass--EXTENSION" for power-up diagnostics.

**Related Commands**

DIAG:RESULT:FLAG?

## DISplay? (Query Only)

Returns the current display settings.

### Group

Display

### Syntax

DISplay?

### Returns

The current display settings

### Examples

DISPLAY?

might return :DISPLAY:FORMAT YT;STYLE VECTORS;  
PERSISTENCE OFF;CONTRAST 50

### Related Commands

None

## DISplay:CONTRast

Sets or queries the contrast of the LCD display. This command is equivalent to setting Contrast in the Display menu.

### Group

Display

### Syntax

DISplay:CONTRast <NR1>

DISplay:CONTRast?

### Arguments

<NR1> is an integer in the range from 1 through 100. The larger the value, the greater the screen contrast.

### Examples

DISplay:CONTRast 63  
sets the display contrast to 63.

### Related Commands

None

## DISplay:FORMat

Sets or queries the oscilloscope display format. This command is equivalent to setting Format in the Display menu.

### Group

Display

### Syntax

DISplay:FORMat { XY | YT }

DISplay:FORMat?

### Arguments

XY displays the voltage of channel 1 (horizontal axis) against the voltage of channel 2 (vertical axis).

---

**NOTE** Setting the display format to XY turns cursors off. Sending the CURSor:FUNCTioN command when the display format is XY causes the oscilloscope to generate event 221 (settings conflict) and leaves the display in XY format.

---

YT sets the display to vertical values versus time format and is the normal mode.



**Examples**

DISPLAY:FORMAT YT

selects a voltage versus time format for the display.

DISPLAY:FORMAT?

might return XY for the display format.

**Related Commands**

None

## DISplay:PERSiStence

Sets the length of time that data points are displayed.

**Group**

Display

**Syntax**

DISplay:PERSiStence { 1 | 2 | 5 | INF | OFF }

DISplay:PERSiStence?

**Arguments**

1 | 2 | 5 specifies the length, in seconds, that the waveform points are displayed on the screen.

INF specifies infinite persistence.

OFF specifies that persistence is turned off.

**Returns**

0 means that the persistence is turned off.

2 or 5 means that the persistence is set to two or five seconds.

99 means that the persistence is set to infinite.

### **Examples**

**DISPLAY:PERSISTENCE 5**

specifies that the waveform points are displayed on the screen for five seconds before they fade.

### **Related Commands**

DISplay:STYLE

## **DISplay:STYLE**

Selects how to display the waveform data. This command is equivalent to setting Type in the Display menu.

### **Group**

Display

### **Syntax**

DISplay:STYLE { DOTs | VECTors }

DISplay:STYLE?

### **Arguments**

DOTs displays individual data points.

VECTors connects adjacent data points.

### **Examples**

**DISPLAY:STYLE VEC**

sets the display to connect adjacent data points.

**DISPLAY:STYLE?**

might return DOTs indicating that the display shows individual waveform data points.

### **Related Commands**

DISplay:PERsistence

## **ERRLOG:FIRST? (Query Only)**

Returns the first entry in the error log, or an empty string if the error log is empty. Use this command along with ERRLOG:NEXT? to retrieve error log messages.

### **Group**

Calibration and Diagnostic

### **Syntax**

ERRLOG:FIRST?

### **Returns**

Refer to the *TDS 200-Series Digital Real-Time Oscilloscope Service Manual* for information about error log message format.

### **Related Commands**

ERRLOG:NEXT?

## **ERRLOG:NEXT? (Query Only)**

Returns the next entry in the error log, or an empty string if the error log is empty or you have reached the end of the log. To start at the top of the error log, run the ERRLOG:FIRST? query to return the first error log message. Then use the ERRLOG:NEXT? query to step through the error log.

### **Group**

Calibration and Diagnostic

### **Syntax**

ERRLOG:NEXT?

### Returns

Refer to the *TDS 200-Series Digital Real-Time Oscilloscope Service Manual* for information about error log message format.

### Related Commands

ERRLOG:FIRST?

## \*ESE

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 page 3–1.

### Group

Status and Error

### Syntax

\*ESE <NR1>

\*ESE?

### Arguments

<NR1> is a value in the range from 0 through 255. The binary bits of the ESER are set according to this value.

The power-on default for ESER is 0 if \*PSC is 1. If \*PSC is 0, the ESER maintains its value through a power cycle.

---

**NOTE** *Setting the DESER and the 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 DESE command to set the DESER. A discussion of event handling begins on page 3–1.*

---

**Examples**

**\*ESE 209**

sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

**\*ESE?**

might return the string **\*ESE 186**, showing that the ESER contains the binary value 10111010.

**Related Commands**

**\*CLS, DESE, \*ESR?, EVENT?, EVMsg? \*SRE, \*STB?**

**\*ESR? (Query Only)**

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 page 3–1.

**Group**

Status and Error

**Syntax**

**\*ESR?**

**Returns**

Contents of the Standard Event Status Register.

**Examples**

**\*ESR?**

might return the value 213, showing that the SESR contains binary 11010101.

**Related Commands**

**ALLEv?, \*CLS, DESE, \*ESE, EVENT?, EVMsg?, \*SRE, \*STB?**

## EVENT? (Query Only)

Returns from the Event Queue an event code that provides information about the results of the last \*ESR? read. EVENT? also removes the returned value from the Event Queue. A discussion of event handling begins on page 3–1.

### Group

Status and Error

### Syntax

EVENT?

### Returns

Event code <NR1>

### Examples

EVENT?

might return the response :EVENT 110, showing that there was an error in a command header.

### Related Commands

ALLEv?, \*CLS, DESE, \*ESE, \*ESR?, EVMsg?, \*SRE, \*STB?

## EVMsg? (Query Only)

Removes from the Event Queue a single event code associated with the results of the last \*ESR? read, and returns the event code along with an explanatory message. A more detailed discussion of event handling begins on page 3–1.

### Group

Status and Error

### 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 as possible is returned without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

**Examples**

EVMSG?

might return the message :EVMSG 110,"Command header error".

**Related Commands**

ALLEv?, \*CLS, DESE, \*ESE, \*ESR?, EVENT?, \*SRE, \*STB?

## EVQtY? (Query Only)

Returns the number of event codes that are in the Event Queue. This is useful when using ALLEv? since it lets you know exactly how many events will be returned.

**Group**

Status and Error

**Syntax**

EVQtY?

**Returns**

<NR1>

### **Examples**

EVQTY?

might return 3 as the number of event codes in the Event Queue.

### **Related Commands**

ALLEv?, EVENT?, EVMsg?

## **FACTory (No Query Form)**

Resets the oscilloscope to its factory default settings. Refer to *Appendix B* for a list of the factory default settings.

### **Group**

Miscellaneous

### **Syntax**

FACTory

Setting the oscilloscope to factory default has the following impact on the programming interface:

- Clears the Event Status Enable Register
- Clears the Service Request Enable Register
- Sets the Device Event Status Enable Register to 255
- Sets the Power On Status Clear Flag to TRUE
- Enables all Command Headers (HEADER ON)
- Sets the macro defined by \*DDT to a “zero-length field”
- Clears the pending operation flag and associated operations
- Performs the equivalent of DATA INIT



The FACTory command does not alter the following:

- The state of the RS-232 or GPIB interface
- The selected GPIB address
- Calibration data that affects device specifications
- Stored settings
- Stored waveforms or data
- Hard copy parameters
- Display messages language selection
- Front panel LOCK state
- Verbose state

**Arguments**

None

**Examples**

See *Appendix B: Factory Setup*.

**Related Commands**

\*PSC, \*RCL, RECALL:SETUp, \*RST, \*SAV, SAVe:SETUp

## HARDCopy

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 front-panel HARDCOPY button.

HARDCopy? returns format, layout, and port information.

---

**NOTE** This command is not IEEE Std 488.2-1987 compatible.

---

### Group

Hard copy

### Syntax

HARDCopy { ABOrt | START }

HARDCopy?

### Arguments

ABOrt terminates the hard copy output in process.

---

**NOTE** DCL does not clear the output queue once a hard copy is in process. The only way to abort the hard copy process is to send the HARDCopy ABOrt command. The output queue can then be cleared by sending a DCL message.

---

START initiates a screen copy that is sent to the port specified by the HARDCOPY:PORT command.

---

**NOTE** Use the \*WAI command between HARDCopy START commands to ensure that the first hard copy is complete before starting another.

---

**Examples**

HARDCOPY ABORT

stops any hard copy output that is in process.

**Related Commands**

\*WAI

**HARDCopy:FORMat**

Sets the hard copy output data format. This command is the same as setting Format in the UTILITY ► Options ► Hard Copy Setup menu.

**Group**

Hard copy

**Syntax**

HARDCopy:FORMat { BMP | DESKJet | EPSOn | LASERJet |  
THInkjet | PCX | EPSIMAGE | INTERLEAF |  
DPU411 | DPU412 }

HARDCopy:FORMat?

**Arguments**

BMP sets the hard copy output data to Microsoft Windows Bitmap format.

DESKjet sets the hard copy output data to high-resolution printer format.

EPSOn sets the hard copy output data to 9-pin or 24-pin dot matrix printer format.

LASERJet sets the hard copy output data to laser printer format.

THInkjet sets the hard copy output data to inkjet printer format.

PCX sets the hard copy output data to DOS Paintbrush format.

EPSIMAGE sets the hard copy output data to Postscript format.

INTERLEAF sets the hard copy output data to Interleaf file format.

DPU411 or DPU412 sets the hard copy output data to Seiko DPU411 or DPU412 printer format.

### Examples

HARDCOPY:FORMAT LASERJet

sets the hard copy output format to LaserJet.

HARDCOPY:FORMAT?

might return EPSON as the hard copy output format.

### Related Commands

None

## HARDCopy:LAYout

Selects the printing orientation. This command is equivalent to setting Layout in the UTILITY ► Options ► Hard Copy Setup menu.

### Group

Hard copy

### Syntax

HARDCopy:LAYout { LANDscape | PORTRait }

HARDCopy:LAYout?

### Arguments

LANDscape specifies that the bottom of the hard copy is along the long side of the page.

PORTRait specifies that the bottom of the hard copy is along the short side of the page. This is the standard format.

### Examples

HARDCOPY:LAYOUT?

might return PORTRAIT as the page layout format of the hard copy output.

**Related Commands**

None

**HARDCopy:PORT**

Selects where to send the hard copy data when the oscilloscope receives the next **HARDCOPY START** command. This command is equivalent to setting Port in the **UTILITY ► Options ► Hard Copy Setup** menu.

**Group**

Hard copy

**Syntax**

**HARDCopy:PORT** { **CEN**tronics | **RS232** | **GP**IB }

**HARDCopy:PORT?**

**Arguments**

**CEN**tronics specifies that the hard copy data is sent out the Centronics port.

**RS232** specifies that the hard copy data is sent out the RS232 port.

**GP**IB specifies that the hard copy data is sent out the GPIB port.

**Examples**

**HARDCOPY:PORT?**

might return **RS232** as the selected hard copy output port.

**Related Commands**

**HARDCopy**

**HDR**

This command is identical to the **HEAD**er query and is included for compatibility with older Tektronix instruments.

**Group**

Miscellaneous

**Syntax**

HDR { <NR1> | OFF | ON }

HDR?

**Arguments**

<NR1> | OFF | ON

Refer to descriptions in the HEADER command.

**Examples**

None

**Related Commands**

HEADer

## HEADer

Sets and queries the Response Header Enable State that causes the oscilloscope to either include or omit headers on query responses. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); they never return headers.

**Group**

Miscellaneous

**Syntax**

HEADer { <NR1> | OFF | ON }

HEADer?

**Arguments**

ON or <NR1>  $\neq 0$  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.

OFF or <NR1> = 0 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.

**Examples**

HEADER OFF

causes the oscilloscope to omit headers from query responses.

HEADER?

might return the value 1, showing that the Response Header Enable State is true.

**Related Commands**

VERBose

**HORizontal? (Query Only)**

Returns all settings for the horizontal commands. The commands HORizontal:MAIn:SCAle, HORizontal:MAIn:SECdiv, HORizontal:SCAle, and HORizontal:SECdiv are equivalent, so HORizontal:MAIn:SCAle is the value that is returned.

The commands HORizontal:MAIn:POSition and HORizontal:POSition are equivalent, so HORizontal:MAIn:POSition is the value that is returned.

**Group**

Horizontal

**Syntax**

HORizontal?

**Returns**

Returns all horizontal settings

**Examples**

HORIZONTAL?

might return the string :HORIZONTAL:VIEW MAIN;RECORDLENGTH 2500;MAIN:POSITION 0.0E0;SCALE 5.0E-4;HORIZONTAL:DELAY:POSITION 0.0E0;SCALE 2.5E-4

**Related Commands**

None

## **HORizontal:DELay? (Query Only)**

Returns all settings for the window time base. The commands HORizontal:DELay:SECdiv and HORizontal:DELay:SCAle are equivalent, so only the values for HORizontal:DELay:SCAle are returned.

### **Group**

Horizontal

### **Syntax**

HORizontal:DELay?

### **Returns**

All settings for the window time base

### **Examples**

HORIZONTAL:DELAY?

might return :HORIZONTAL:DELAY:POSITION 0.0E0;  
SCALE 5.0E-4.

## **HORizontal:DELay:POSition**

Sets or queries the window time base horizontal position. This command is equivalent to adjusting the Horizontal Position when Window Zone or Window is selected from the Horizontal menu.

### **Group**

Horizontal

### **Syntax**

HORizontal:DELay:POSition <NR3>

HORizontal:DELay:POSition?

### **Arguments**

<NR3> is the position in seconds. This value is the difference between the trigger point and the center graticule. Positive values place the trigger before the center graticule.



**Examples**

HORIZONTAL:DELAY:POSITION 2.0E-6

sets the window position to 2  $\mu$ s before the center graticule.

HORIZONTAL:DELAY:POSITION?

might return -1.0E-3, indicating that the window position is 1 ms after the center graticule.

**Related Commands**

HORizontal:DELay:SECdiv

## **HORizontal:DELay:SCALE**

Sets the time per division for the oscilloscope window time base.

This command is equivalent to setting SEC/DIV when Window Zone or Window is selected from the Horizontal menu.

**Group**

Horizontal

**Syntax**

HORizontal:DELay:SCALE <NR3>

HORizontal:DELay:SCALE?

**Arguments**

<NR3> is the time per division. The range is 5 ns to 5 s in a 1-2.5-5 sequence. If the window time base scale is set slower than the main time base scale, both the main and window time base scales are set to the window scale value.

**Examples**

HORIZONTAL:DELAY:SCALE 2.0E-6

sets the window scale to 2  $\mu$ s per division.

HORIZONTAL:DELAY:SCALE 9.0E-6

sets the window scale to 10  $\mu$ s per division. Since 9  $\mu$ s is not a correct value within the 1-2.5-5 sequence, it is automatically set to the closest correct value.

`HORIZONTAL:DELAY:SCALE?`

might return `1.0E-3`, indicating that the window scale is 1 ms per division.

**Related Commands**

`HORizontal:DELay:SECdiv`

## **`HORizontal:DELay:SECdiv`**

This command is identical to the `HORizontal:DELay:SCAle` command. It is provided to maintain program compatibility with some older models of Tektronix oscilloscopes.

**Group**

Horizontal

**Syntax**

`HORizontal:DELay:SECdiv <NR3>`

`HORizontal:DELay:SECdiv?`

**Arguments**

Same as `HORizontal:DELay:SCAle`

**Examples**

Same as `HORizontal:DELay:SCAle`

**Related Commands**

`HORizontal:DELay:SCAle`

## **HORizontal:MAIn? (Query Only)**

Returns all settings for the oscilloscope main time base. The commands HORizontal:MAIn:SECdiv and HORizontal:MAIn:SCALE are identical so only HORizontal:MAIn:SCALE is returned.

### **Group**

Horizontal

### **Syntax**

HORizontal:MAIn?

### **Returns**

All settings for the main time base.

### **Examples**

HORIZONTAL:MAIN?

might return :HORIZONTAL:MAIN:POSITION 0.0E0;SCALE 5.0E-4.

### **Related Commands**

HORizontal:SCALE, HORizontal:SECdiv, HORizontal:MAIn:SECdiv

## **HORizontal:MAIn:POSition**

Sets or queries the main time base horizontal position. This command is equivalent to adjusting the Horizontal Position when Main is selected from the Horizontal menu.

### **Group**

Horizontal

### **Syntax**

HORizontal:MAIn:POSition <NR3>

HORizontal:MAIn:POSition?

### Arguments

<NR3> is the position in seconds. This value is the difference between the trigger point and the center graticule. Positive values place the trigger before the center graticule.

### Examples

HORIZONTAL:MAIn:POSition 2.0E-6

sets the main trigger position to 2  $\mu$ s before the center graticule.

HORIZONTAL:MAIn:POSition?

might return -1.0E-3, indicating that the main trigger position is 1 ms after the center graticule.

### Related Commands

HORizontal:DELAy:POSition

## HORizontal:MAIn:SCALE

Sets the time per division for the main time base. This command is equivalent to setting SEC/DIV when Main is selected from the Horizontal menu.

### Group

Horizontal

### Syntax

HORizontal:MAIn:SCALE <NR3>

HORizontal:MAIn:SCALE?

### Arguments

<NR3> is the time per division. The range is 5 ns to 5 s in a 1-2.5-5 sequence.

### Examples

HORIZONTAL:MAIN:SCALE 2.5E-6

sets the main scale to 2.5  $\mu$ s per division.

**Related Commands**

HORizontal:DELay:SCAle, HORizontal:DELay:SECdiv, HORizontal:MAIn:SECdiv

**HORizontal:MAIn:SECdiv**

Sets the time per division for the oscilloscope main time base. This command is identical to the HORizontal:MAIn:SCAle command. It is provided to maintain program compatibility with some older models of Tektronix oscilloscopes.

**Group**

Horizontal

**Syntax**

HORizontal:MAIn:SECdiv <NR3>

HORizontal:MAIn:SECdiv?

**Arguments**

<NR3> is the time per division. The range is 5 ns to 5 s in a 1-2.5-5 sequence. Values that are not in a 1-2.5-5 sequence are set to the closest valid value.

**Examples**

HORIZONTAL:MAIN:SECdiv 2.5E-6  
sets the main scale to 2.5  $\mu$ s per division.

**Related Commands**

HORizontal:DELay:SCAle, HORizontal:DELay:SECdiv, HORizontal:MAIn:SCAle

## **HORizontal:POSition**

Sets or queries the main time base horizontal position. This command is identical to the HORizontal:MAIn:POSition command. This command is included for compatibility.

### **Group**

Horizontal

### **Syntax**

HORizontal:POSition <NR3>

HORizontal:POSition?

### **Arguments**

<NR3> is the position in seconds. This value is the difference between the trigger point and the center graticule. Positive values place the trigger before the center graticule.

### **Examples**

HORIZONTAL:POSition 2.0E-6

sets the main trigger position to 2  $\mu$ s before the center graticule.

HORIZONTAL:POSition?

might return -1.0E-3, indicating that the main trigger position is 1 ms after the center graticule.

### **Related Commands**

HORizontal:DElay:POSition, HORizontal:MAIn:POSition

## **HORizontal:RECOrdlength? (Query Only)**

Returns the number of acquisition data points. For TDS 200-series oscilloscopes this value is always 2500. This command is provided to maintain program compatibility with other TDS-series oscilloscopes.

### **Group**

Horizontal

### **Syntax**

**HORizontal:RECOrdlength?**

### **Returns**

2500

### **Examples**

**HORIZONTAL:RECOrdlength?**

returns 2500 as the number of data points in each record.

### **Related Commands**

**WFMPre:NR\_Pt?**, **WFMPre:<wfm>:NR\_Pt?**

## **HORizontal:SCALE**

Sets the time per division for the main time base and is identical to the **HORizontal:MAIn:SCALE** command. It is included for compatibility purposes.

### **Group**

Horizontal

### **Syntax**

**HORizontal:SCALE <NR3>**

**HORizontal:SCALE?**

### **Arguments**

**<NR3>** is the time per division. The range is 5 ns to 5 s in a 1-2.5-5 sequence.

**Examples**

HORIZONTAL:SCALE 2.5E-6  
sets the main scale to 2.5  $\mu$ s per division.

**Related Commands**

HORizontal:MAIn:SCAle

## **HORizontal:SECdiv**

Sets the time per division for the main time base and is identical to the HORizontal:MAIn:SCAle command. It is included for compatibility purposes.

**Group**

Horizontal

**Syntax**

HORizontal:SECdiv <NR3>

HORizontal:SECdiv?

**Arguments**

<NR3> is the time per division. The range is 5 ns to 5 s in a 1-2.5-5 sequence.

**Examples**

HORIZONTAL:SECdiv 2.5E-6  
sets the main scale to 2.5  $\mu$ s per division.

**Related Commands**

HORizontal:MAIn:SCAle



## **HORizontal:VIEW**

Specifies whether the horizontal display uses the main, window zone, or window view. This is equivalent to setting the View in the Horizontal menu.

### **Group**

Horizontal

### **Syntax**

**HORizontal:VIEW** { MAIN | WINDow | ZONE }

**HORizontal:VIEW?**

### **Arguments**

**MAIN** specifies that the waveform is horizontally scaled relative to the main time base.

**WINDow** specifies that the waveform is horizontally scaled relative to the window (delay) time base.

**ZONE** specifies that the portion of the waveform that is within the vertical cursor bars is acquired through the window (delay) time base.

### **Examples**

**HORIZONTAL:VIEW MAIN**

specifies that the waveform is horizontally scaled relative to the main time base.

## ID? (Query Only)

Returns identifying information about the instrument and its firmware in Tektronix Codes and Formats notation.

---

**NOTE** *ID? must be the last command when it is part of a concatenated statement. Otherwise the oscilloscope generates event message 440.*

---

### Group

Status and Events

### Syntax

ID?

### Returns

The instrument identification in the following format:

TEK/<model number>,CF:91.1CT,FV:v<oscilloscope firmware version number>,TDS2XX:YYV:v<module firmware version number>

Where XX is oscilloscope model and YY is the module type (CM for Communications Module or MM for Measurement Module).

The \*IDN? and ID? responses are slightly different.

### Examples

ID?

might return the response

ID TEK/TDS 220,CF:91.1CT,FV:v1.00 TDS2CM:CMV:v1.00

### Related Commands

\*IDN?

## **\*IDN? (Query Only)**

Returns the oscilloscope identification code in IEEE 488.2 notation.

---

**NOTE** *IDN? must be the last command when it is part of a concatenated statement. Otherwise the oscilloscope generates event message 440.*

---

### **Group**

Status and Events

### **Syntax**

\*IDN?

### **Returns**

The instrument identification in the following format:

TEKTRONIX,<model number>,0,CF:91.1CT FV:v<oscilloscope  
firmware version number> TDS2XX:YYV:v<module firmware  
version number>

Where XX is oscilloscope model and YY is the module type (CM for Communications Module or MM for Measurement Module).

The \*IDN? and ID? responses are slightly different.

### **Examples**

\*IDN?

might return the response

TEKTRONIX,TDS 220,0,CF:91.1CT FV:v1.00 TDS2CM:CMV:v1.00

### **Related Commands**

ID?

## LANGuage

Sets or queries the languages that the oscilloscope uses to display information on the screen. This is equivalent to the Language selection in the Utility menu.

### Group

Miscellaneous

### Syntax

```
LANGuage { ENGLISH | FRENch | GERMan | ITALian |  
PORTUGuese | SPANish | JAPANese | KOREan |  
TRADitionalchinese | SIMPlifiedchinese }
```

LANGuage?

### Arguments

The argument specifies which language will be used to display oscilloscope information on the screen.

### Examples

LANGuage FRENch  
specifies that the oscilloscope display information in French.

LANGuage?  
might return SPANish.

### Related Commands

None

## LOCK

Enables and disables all front-panel buttons and knobs. There is no front-panel equivalent.

### Group

Miscellaneous

### Syntax

```
LOCK { ALL | NONE }
```

LOCK?

### Arguments

ALL disables all front-panel controls.

NONE enables all front-panel controls. This is equivalent to the UNLOCK ALL command.

### Examples

```
LOCK ALL
```

locks the front-panel controls.

```
LOCK?
```

returns NONE when the front-panel controls are enabled by this command.

### Related Commands

UNLOCK

## **\*LRN? (Query Only)**

Returns a string listing the oscilloscope settings, except for calibration configuration information. You can use this string to return the oscilloscope to the state it was in when you sent the \*LRN? command.

### **Group**

Miscellaneous

### **Syntax**

\*LRN?

---

**NOTE** *\*LRN? always returns a string including command headers, regardless of the setting of the HEADer command. This is because the returned string is intended to be sent back to the oscilloscope as a command string. The VERBose command can still be used normally to specify whether the returned headers should be abbreviated.*

---

### **Returns**

None

### **Examples**

\*LRN?

a partial response might look like this:

```
:ACQUIRE:STATE 1;MODE SAMPLE;NUMAVG 16;  
STOPAFTER RUNSTOP;STATE 1;:HEADER 1;:VERBOSE 1;  
:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;POSITION1  
1.00E-6;POSITION2 9.00E-6
```

### **Related Commands**

HEADer, SET?, VERBose

## MATH? (Query Only)

Returns the definition for the math waveform.

### Group

Vertical

### Syntax

MATH?

### Returns

Definition for the math waveform.

### Examples

MATH?

returns :MATH:DEFINE "CH1 + CH2" if the math waveform is defined as channel 1 plus channel 2.

### Related Commands

MATH:DEFINE

## MATH:DEFINE

Performs the specified mathematical operation on the input signal or signals. To turn the math waveform on or off, use the SElect:<wfm> command (page 2–138).

### Group

Vertical

### Syntax

MATH:DEFINE <QString>

MATH:DEFINE?

### Arguments

<QString> specifies a math waveform, and can be one of the following:

TDS210/TDS220 (firmware below V 2.00) with TDS2CM:

CH1-CH2

CH2-CH1

CH1+CH2

-CH1

-CH2

TDS210/TDS220 (firmware below V 2.00) with TDS2MM:

CH1+CH2

(to subtract, invert one of the waveforms via CH<x>:INVert, and then add)

FFT (CH<x> [, <window>])

TDS210/TDS220 (firmware V 2.00 and above) with TDS2CM:

CH1+CH2

CH1-CH2

CH2-CH1

TDS210/TDS220 (firmware V 2.00 and above) with TDS2MM:

CH1+CH2

CH1-CH2

CH2-CH1

FFT (CH<x> [, <window>])

TDS224 with TDS2CM:

CH1+CH2

CH3+CH4

CH1-CH2

CH2-CH1

CH3-CH4

CH4-CH3

TDS224 with TDS2MM:

CH1+CH2

CH3+CH4

CH1-CH2

CH2-CH1

CH3-CH4

CH4-CH3

FFT (CH<x> [, <window>])



For FFT, the optional <window> argument is HANning, FLATtop, or RECTangular.

Remember that <QString> must be enclosed in quotes. You can use white space characters between words.

### **Examples**

MATH:DEFINE "CH1+CH2"

sets the math waveform so that it displays the sum of channel 1 and channel 2.

MATH:DEFINE "FFT (CH1, HANNING)"

sets the math waveform to display an FFT of channel 1 using a Hanning window (TDS2MM Only).

### **Related Commands**

MATH?

## **MATH:FFT:HORizontal:POSition (TDS2MM Only)**

Sets or queries the FFT math waveform horizontal position.

### **Group**

Horizontal

### **Syntax**

MATH:FFT:HORizontal:POSition <NR3>

MATH:FFT:HORizontal:POSition?

### **Arguments**

<NR3> specifies the point in the FFT waveform data record to display at the center vertical graticule line. The data record point is a percentage of the total record length, in the range of 0 to 100. The default value is 50. How much data is displayed depends on the FFT Zoom factor setting. The oscilloscope rounds <NR3> to the nearest workable value.

### Examples

MATH:FFT:HORIZONTAL:POSITION 25

sets the FFT waveform horizontal position such that the data at the 25% point of the record is centered on the display.

MATH:FFT:HORIZONTAL:POSITION?

might return 5.0E1, indicating that the 50% point in the data record is horizontally centered on the display.

### Related Commands

MATH:FFT:HORizontal:SCAle

## MATH:FFT:HORizontal:SCAle (TDS2MM Only)

Sets or queries the FFT math waveform horizontal zoom factor.

### Group

Horizontal

### Syntax

MATH:FFT:HORizontal:SCAle <NR3>

MATH:FFT:HORizontal:SCAle?

### Arguments

<NR3> specifies the horizontal magnification factor, where the axis of magnification is the center vertical graticule line. Valid zoom factors are 1, 2, 5, and 10. If other values are entered the oscilloscope rounds the value of <NR3> to the nearest valid zoom factor.

### Examples

MATH:FFT:HORIZONTAL:SCALE 5

sets the FFT waveform horizontal zoom factor to X5.

MATH:FFT:HORIZONTAL:SCALE?

might return 2.0E0, indicating that the zoom factor setting is X2.

### Related Commands

MATH:FFT:HORizontal:POSition

## **MATH:FFT:VERTical:POStion (TDS2MM Only)**

Sets or queries the FFT math waveform vertical position.

### **Group**

Vertical

### **Syntax**

MATH:FFT:VERTical:POStion <NR3>

MATH:FFT:VERTical:POStion?

### **Arguments**

<NR3> specifies the FFT waveform vertical position in divisions.

### **Examples**

MATH:FFT:VERTICAL:POSITION 2

sets the FFT waveform vertical position to 2 major divisions above the center horizontal graticule line.

MATH:FFT:VERTICAL:POSITION?

might return 5, indicating that the FFT waveform 0 reference point is 5 major divisions from the center horizontal graticule line.

### **Related Commands**

MATH:FFT:VERTical:SCAle

## **MATH:FFT:VERTical:SCAle (TDS2MM Only)**

Sets or queries the FFT math waveform vertical zoom factor.

### **Group**

Vertical

### **Syntax**

MATH:FFT:VERTical:SCAle <NR3>

MATH:FFT:VERTical:SCAle?

### Arguments

<NR3> specifies the vertical zoom factor, where the axis of magnification is the center horizontal graticule line. Valid zoom factors are 0.5, 1, 2, 5, and 10. If other values are entered the oscilloscope rounds the value of <NR3> to the nearest valid zoom factor.

### Examples

MATH:FFT:VERTICAL:SCALE 5  
sets the FFT waveform vertical zoom factor to X5.

MATH:FFT:VERTICAL:SCALE?  
might return 2.0E0, indicating that the zoom factor setting is X2.

### Related Commands

MATH:FFT:VERTical:POSition

## MEASUREMENT? (Query Only)

Returns all oscilloscope measurement parameters.

### Group

Measurement

### Syntax

MEASUREMENT?

### Returns

Oscilloscope measurement parameters

### Examples

MEASUREMENT?  
might return the following:

```
:MEASUREMENT:MEAS1:TYPE PERIOD;UNITS "s";SOURCE CH1;  
:MEASUREMENT:MEAS2:TYPE FREQUENCY;UNITS "Hz";SOURCE CH1;  
:MEASUREMENT:MEAS3:TYPE PK2PK;UNITS "V";SOURCE CH2;  
:MEASUREMENT:MEAS4:TYPE MEAN;UNITS "V";SOURCE CH2;  
:MEASUREMENT:IMMED:TYPE CRMS;UNITS "V";SOURCE CH1
```

**Related Commands**

None

**MEASUrement:IMMed? (Query Only)**

Returns all immediate measurement setup parameters. Immediate queries and commands are the preferred methods for programming. An immediate measurement selection is not visible or accessible through the display screen or front panel.

**Group**

Measurement

**Syntax**

MEASUrement:IMMed?

**Returns**

Immediate measurement setup parameters

**Examples**

MEASUREMENT:IMMED?  
might return PERIOD;"s";CH1

**Related Commands**

None

**MEASUrement:IMMed:SOURce**

Sets or queries the source for all immediate measurements.

**Group**

Measurement

**Syntax**

MEASUrement:IMMed:SOURce CH<x>

MEASUrement:IMMed:SOURce?

### **Arguments**

CH<x> is an input channel.

### **Examples**

MEASUREMENT:IMMED:SOURCE CH1

specifies channel 1 as the immediate measurement source.

### **Related Commands**

None

## MEASUrement:IMMed:TYPe

Sets or queries the immediate measurement type.

### Group

Measurement

### Syntax

```
MEASUrement:IMMed:TYPe { FREQuency | MEAN | PERIOd |  
PK2pk | CRMs | RISE | FALL | PWidth | NWidth }
```

```
MEASUrement:IMMed:TYPe?
```

### Arguments

FREQuency is the reciprocal of the period measured in Hertz.

MEAN is the arithmetic mean over the entire waveform.

PERIOd is the time, in seconds, it takes for one complete signal cycle to happen.

PK2pk is the absolute difference between the maximum and minimum amplitude.

CRMs is the true Root Mean Square voltage over one cycle.

RISe (TDS2MM Only) is the rise time between 10% and 90% of the first rising edge of the waveform. Rising edge must be displayed to measure. The TDS2MM automatically calculates the 10% and 90% measurement points.

FALL (TDS2MM Only) is the fall time between 90% and 10% of the first falling edge of the waveform. Falling edge must be displayed to measure. The TDS2MM automatically calculates the 10% and 90% measurement points.

PWidth (TDS2MM Only) is the positive pulse width between the first rising edge and the next falling edge at the waveform 50% level. Rising and falling edges must be displayed to measure. The TDS2MM automatically calculates the 50% measurement point.

NWIdth (TDS2MM Only) is the negative pulse width between the first falling edge and the next rising edge at the waveform 50% level. Falling and rising edges must be displayed to measure. The TDS2MM automatically calculates the 50% measurement point.

**Examples**

MEASUREMENT:IMMED:TYPE FREQUENCY

defines the immediate measurement to be a frequency measurement.

**Related Commands**

None

## **MEASUREMENT:IMMED:UNITS? (Query Only)**

Returns the units for the immediate oscilloscope measurement.

**Group**

Measurement

**Syntax**

MEASUREMENT:IMMED:UNITS?

**Returns**

<QString> returns "V" for volts, "s" for seconds, or "Hz" for hertz.

**Examples**

MEASUREMENT:IMMED:UNITS?

might return "s", indicating that the unit for the immediate measurement is seconds.

**Related Commands**

MEASUREMENT:IMMED:TYPE



## MEASUrement:IMMed:VALue? (Query Only)

Executes the immediate oscilloscope measurement specified by the MEASUrement:IMMed:TYPE command. The measurement is taken on the source specified by the MEASUrement:IMMed:SOURce command. Immediate queries and commands are the preferred measurement method for programming.

In order to see if the measurement was successful, use the \*ESR? and EVENT? commands to read the Standard Event Status Register (SESR). If the channel specified by MEASUrement:IMMed:SOURce is not currently displayed, the oscilloscope generates event 2225 and returns 9.9E37.

---

**NOTE** *If trigger view is active, scan mode is in effect, or the display format set to XY, this query returns 9.9E37 and generates event 221 (Settings conflict).*

---

### Group

Measurement

### Syntax

MEASUrement:IMMed:VALue?

### Returns

<NR3>

### Examples

MEASUrement:IMMed:VALue?

might return 28.75E6 if you are measuring the frequency of a 28.75 MHz signal.

MEASUrement:IMMed:VALue?

might return 9.9E37, which means that the channel specified by MEASUrement:IMMed:SOURce is not currently displayed.

### Related Commands

None

## **MEASUrement:MEAS<x>? (Query Only)**

Returns all measurement parameters for the displayed oscilloscope periodic measurement specified by <x>, where <x> is 1 through 4.

### **Group**

Measurement

### **Syntax**

MEASUrement:MEAS<x>?

### **Returns**

Settings for the specified measurement source.

### **Examples**

MEASUREMENT:MEAS3?  
might return PERIOD;"s";CH1

### **Related Commands**

None

## **MEASUrement:MEAS<x>:SOUrce**

Sets or queries the source for the measurement. This is equivalent to selecting the measurement source in the MEASURE menu.

### **Group**

Measurement

### **Syntax**

MEASUrement:MEAS<x>:SOUrce CH<y>

MEASUrement:MEAS<x>:SOUrce?

### **Arguments**

CH<y> specifies the measurement source channel.

**Examples**

MEASUREMENT:MEAS2:SOURCE CH1

Sets the source of MEAS2 to channel 1.

**Related Commands**

None

## **MEASUREMENT:MEAS<x>:TYPE**

Sets or queries the oscilloscope measurement type for the measurement specified by <x>. This is equivalent to selecting the measurement type in the MEASURE menu. Setting the type to anything other than NONE displays the MEASURE menu on the screen.

**Group**

Measurement

**Syntax**

MEASUREMENT:MEAS<x>:TYPE { FREQUENCY | MEAN | PERIOD | PK2pk | CRMS | RISE | FALL | PWIDTH | NWIDTH | NONE }

MEASUREMENT:MEAS<x>:TYPE?

**Arguments**

FREQUENCY is the reciprocal of the period measured in Hertz.

MEAN is the arithmetic mean over the entire waveform.

PERIOD is the time, in seconds, it takes for one complete signal cycle to happen.

PK2pk is the absolute difference between the maximum and minimum signal amplitude.

CRMS is the true Root Mean Square value over one waveform cycle.

RISE (TDS2MM Only) is the rise time between 10% and 90% of the first rising edge of the waveform. Rising edge must be displayed to measure. The TDS2MM automatically calculates the 10% and 90% measurement points.

**FALL** (TDS2MM Only) is the fall time between 90% and 10% of the first falling edge of the waveform. Falling edge must be displayed to measure. The TDS2MM automatically calculates the 10% and 90% measurement points.

**PWIdth** (TDS2MM Only) is the positive pulse width between the first rising edge and the next falling edge at the waveform 50% level. Rising and falling edges must be displayed to measure. The TDS2MM automatically calculates the 50% measurement point.

**NWIdth** (TDS2MM Only) is the negative pulse width between the first falling edge and the next rising edge at the waveform 50% level. Falling and rising edges must be displayed to measure. The TDS2MM automatically calculates the 50% measurement point.

**NONE** disables the measurement specified by <x>.

### Examples

MEASUREMENT:MEAS3:TYPE CRMS

specifies MEAS3 to calculate the true Root Mean Square value.

### Related Commands

None

## MEASUREMENT:MEAS<x>:UNITS? (Query Only)

Returns the units for the oscilloscope measurement specified by MEASUREMENT:MEAS<x>:TYPE.

### Group

Measurement

### Syntax

MEASUREMENT:MEAS<x>:UNITS?

### Returns

<QString> returns "V" for volts, "s" for seconds, or "Hz" for hertz.

**Examples**

MEASUREMENT:MEAS3:UNITS?

might return "V", indicating the units for measurement 3 are volts.

**Related Commands**

None

**MEASUREMENT:MEAS<x>:VALUE? (Query Only)**

Returns the value that has been calculated for the oscilloscope measurement specified by <x>.

This value is a display value and will be updated about every 1/2 second if both the MEASURE menu and the MEAS<x> source channel are displayed. If you are acquiring at a slow sweep rate, the oscilloscope may take longer than 1/2 second to update. MEASUREMENT:IMMED is usually more useful than MEASUREMENT:MEAS<x> when using the oscilloscope with external devices.

If MEASUREMENT:MEAS<x>:TYPE is set to NONE, MEASUREMENT:MEAS<x>:VALUE? generates event message 2231 (Measurement error, measurement is not turned on) and returns 9.9E37.

If the channel specified by MEASUREMENT:MEAS<x>:SOURCE is not displayed on the oscilloscope, the oscilloscope generates event message 2225 (Measurement error, no waveform to measure) and returns 9.9E37.

---

**NOTE** *If trigger view is active, scan mode is in effect, or the display format set to XY, this query generates event 221 (Settings conflict) and returns 9.9E37.*

---

**Group**

Measurement

**Syntax**

MEASUREMENT:MEAS<x>:VALUE?

**Returns**

<NR3>

**Examples**

MEASUrement:MEAS3:VALue?  
might return 28.75E6 if measurement number three is frequency.

**Related Commands**

None

**\*OPC**

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.

The \*OPC command lets you synchronize the operation of the oscilloscope with your application program. Synchronization methods are described starting on page 3–10.

For a complete discussion of the use of these registers and the output queue, refer to page 3–1. Table 2–21 lists commands that generate an operation complete message.

**Table 2–21: Commands that generate an Operation Complete message**

Command	Operation
*CAL?, CALibrate:CONTINUE, CALibrate:FACTory, CALibrate:INTERNAL	Internal self-calibration
ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQUENCE)	Busy single sequence acquisition
HARDCopy START	Busy hardcopy output

**Group**

Status and Error

**Syntax**

\*OPC

\*OPC?

**Arguments**

None

**Examples**

\*OPC?

might return a 1, which indicates that the operation is complete.

**Related Commands**

BUSY?, \*WAI

**\*PSC**

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. For a complete discussion of the use of these registers, see page 3–1.

**Group**

Status and Error

**Syntax**

\*PSC <NR1>

\*PSC?

### Arguments

<NR1> = 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.

<NR1>  $\neq$  0 sets the power-on status clear flag true. Sending \*PSC 1, therefore, enables the power-on status clear and prevents any SRQ assertion after power on.

### Examples

\*PSC 0

sets the power-on status clear flag to false.

\*PSC?

might return the value 1, showing that the power-on status clear flag is set to true.

### Related Commands

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

## \*RCL (No Query Form)

Restores the state of the oscilloscope from a copy of its settings stored in memory. (The settings are stored using the \*SAV command.) This command is equivalent to RECALL:SETUp, and performs the same function as the Recall Saved Setup item in the front-panel Save/Recall Setup menu.

### Group

Save and Recall

### Syntax

\*RCL <NR1>

### Arguments

<NR1> is an integer value in the range from 1 to 5, and specifies a setup storage location.



**Examples**

\*RCL 3

restores the oscilloscope from a copy of the settings stored in memory location 3.

**Related Commands**

FACTory, \*LRN?, RECAIl:SETUp, \*RST, \*SAV, SAVe:SETUp

## **RECAIl:SETUp (No Query Form)**

Restores a stored or factory front-panel setup of the oscilloscope. This command is equivalent to selecting Recall Setup or Recall Factory in the Save/Recall Setup menu.

**Group**

Save and Recall

**Syntax**

RECAIl:SETUp { FACTory | <NR1> }

**Arguments**

FACTory selects the factory setup.

<NR1> is a value in the range from 1 to 5 and specifies a setup storage location.

**Examples**

RECALL:SETUP FACTORY

recalls the front-panel setup to its factory defaults.

**Related Commands**

FACTory, \*RCL, \*RST, \*SAV, SAVe:SETUp

## REM (No Query Form)

Specifies a comment. This line is ignored by the instrument.

### Group

Miscellaneous

### Syntax

REM <QString>

### Arguments

<QString> is a string that can have a maximum of 80 characters.

### Examples

REM "This is a comment"  
is ignored by the instrument.

### Related Commands

None

## RS232? (Query Only)

Queries the RS232 settings.

### Group

RS-232

### Syntax

RS232?

### Returns

The current RS-232 settings

### Examples

RS232?  
might return: RS232:BAUD 9600; SOFTFLAGGING 0;  
HARDFLAGGING 1; PARITY NONE; TRANSMIT:TERMINATOR LF

**Related Commands**

RS232: BAUD, RS232: HARDFLagging, RS232: PARity,  
RS232:SOFTFLagging

**RS232:BAUD**

Sets or queries the RS-232-C interface transmission speed. If no flow control (flagging) is used, commands may be received faster than the oscilloscope can process them. Also, if another command is sent immediately after this command, without waiting for the baud rate to be programmed, the first couple of characters may be lost.

**Group**

RS-232

**Syntax**

RS232:BAUD <NR1>

RS232:BAUD?

**Arguments**

<NR1> where <NR1> can be 300, 600, 1200, 2400, 4800, 9600, or 19200.

**Examples**

RS232:BAUD 9600

sets the transmission rate to 9600 baud.

**Related Commands**

RS232: HARDFLagging, RS232: PARity, RS232:SOFTFLagging,  
RS232?

## RS232:HARDFlagging

Sets or queries the state of RS232 hard flagging. When hard flagging is enabled, the instrument sends data as long as CTS (Clear To Send) is asserted. When receiving data, the instrument asserts RTS (Request To Send) until the input buffer is almost full. When the instrument no longer asserts RTS, it continues to read incoming data until the input buffer is full and then reports an input overrun error. The instrument asserts DTR (Data Terminal Ready) at all times when the instrument power is on.

Hard flagging and soft flagging cannot be on at the same time. However, hard and soft flagging can both be off at the same time. Turning hard flagging on turns soft flagging off. Turning soft flagging on turns hard flagging off.

### Group

RS-232

### Syntax

RS232:HARDFlagging { ON | OFF | <NR1> }

RS232:HARDFlagging?

### Arguments

<ON> or <NR1>  $\neq$  0 turns on hard flagging and turns off soft flagging.

<OFF> or <NR1> = 0 turns off hard flagging (RTS always asserted).

### Examples

RS232:HARDFLAGGING ON

turns on hard flagging and turns off soft flagging.

### Related Commands

RS232: BAUD, RS232: PARity, RS232:SOFTFLagging, RS232?

## RS232:PARity

Sets or queries the parity used for all RS-232-C data transfers. When parity is odd or even, the oscilloscope generates the selected parity on output and checks all input against the selected parity. When parity is none, the oscilloscope performs no input parity error checks and generates no output parity.

When the parity (ninth) bit does not match the parity type, the instrument reports a parity error. If another command is sent immediately after this command, without waiting for the parity to be programmed, the first few characters may be lost.

### Group

RS-232

### Syntax

RS232:PARity { EVEN | ODD | NONE }

RS232:PARity?

### Arguments

<EVEN> sets even parity.

<ODD> sets odd parity.

<NONE> sets no parity (no ninth bit transmitted).

### Examples

RS232:PARITY EVEN  
sets even parity.

### Related Commands

RS232: BAUD, RS232: HARDFLagging, RS232:SOFTFLagging,  
RS232?

## RS232:SOFTFlagging

Sets or queries the input and output soft flagging over the RS-232-C port. After receiving an XOFF (DC3), the instrument sends two or less characters. The oscilloscope begins transmitting data again when it receives an XON (DC1) character.

The instrument sends an XOFF character when its input buffer is running out of space. After sending an XOFF character it can receive at least 20 more bytes. It sends XON when its input buffer has an acceptable number of free bytes.

When soft flagging is enabled and binary data is transferred, data transmission will lock up if the data contains XOFF or XON characters.

Hard flagging and soft flagging cannot be on at the same time. However, hard and soft flagging can both be off at the same time. Turning soft flagging on turns hard flagging off. Turning hard flagging on turns soft flagging off.

### Group

RS-232

### Syntax

RS232:SOFTFlagging { ON | OFF | <NR1> }

RS232:SOFTFlagging?

### Arguments

<ON> or <NR1>  $\neq$  0 turns on soft flagging and turns off hard flagging.

<OFF> or <NR1> = 0 turns off soft flagging.

### Examples

RS232:SOFTFLAGGING ON

turns on soft flagging and turns off hard flagging.

### Related Commands

RS232: BAUD, RS232: HARDFlagging, RS232: PARity, RS232?

## **RS232:TRANsmit:TERMinator**

Sets or queries the end-of-line (EOL) terminator. When transmitting, the instrument appends the terminator to the end of each message. When receiving, the instrument accepts all four terminators, regardless of the currently selected terminator. When a combination of multiple characters is selected (CRLF or LFCR), the instrument interprets the first character as the terminator; it treats the second character as a null command.

CR represents an ASCII carriage return character (0x0D) and LF represents an ASCII linefeed character (0x0A).

### **Group**

RS-232

### **Syntax**

RS232:TRANsmit:TERMinator { CR | LF | CRLf | LFCr }

RS232:TRANsmit:TERMinator?

### **Arguments**

<CR> selects the carriage return character as the EOL terminator.

<LF> selects the line feed character as the EOL terminator.

<CRLf> selects the carriage return and line feed characters as the EOL terminator.

<LFCr> selects the line feed and carriage return characters as the EOL terminator.

### **Examples**

RS232:TRANsmit:TERMinator CR  
sets the carriage return as the EOL terminator.

### **Related Commands**

RS232: HARDFLagging, RS232:SOFTFLagging, RS232?

## **\*RST (No Query Form)**

(Reset) Returns the oscilloscope to a known set of instrument settings, but does not purge any stored settings. This command executes a subset of the FACTory command.

### **Group**

Status and Error

### **Syntax**

\*RST

Sending the \*RST command does the following:

- Returns the instrument settings to the factory defaults (refer to *Appendix B*)
- Sets the macro defined by \*DDT to a zero-length field
- Clears the pending operation flag and associated operations

The \*RST command does not alter the following items:

- The state of the RS-232 or GPIB interface
- Calibration data that affects device specifications
- The selected GPIB oscilloscope address
- Stored settings or waveforms
- The Output queue
- The Service Request Enable Register settings
- The Standard Event Status Enable Register settings
- The Power-On Status Clear flag setting
- Hard copy parameters
- Response Header Enable State
- Front panel LOCK state
- Verbose State



**Examples**

None

**Related Commands**

FACTory, \*PSC, \*RCL, RECAIl:SETUp, \*SAV, SAVe:SETUp

**\*SAV (No Query Form)**

(Save) Stores the state of the oscilloscope into a specified memory location. You can later use the \*RCL command to restore the oscilloscope to this saved state. This is equivalent to selecting Save Setup in the Save/Recall Setup menu.

**Group**

Save and Recall

**Syntax**

\*SAV <NR1>

**Arguments**

<NR1> is an integer value in the range from 1 to 5 and specifies a memory location. Any settings that have been stored previously at this location are overwritten.

**Examples**

\*SAV 2

saves the current settings in memory location 2.

**Related Commands**

FACTory, \*RCL, RECAIl:SETUp, SAVe:SETUp

## **SAVe:SETUp (No Query Form)**

Saves the current state of the oscilloscope into the specified memory location. This is equivalent to selecting Save Setup in the Save/Recall Setup menu.

### **Group**

Save and Recall

### **Syntax**

SAVe:SETUp <NR1>

### **Arguments**

<NR1> is an integer value in the range from 1 to 5 and specifies a memory location. Any settings that have been stored previously at this location are overwritten.

### **Examples**

SAVe:SETUp 5

saves the current front-panel setup to memory location 5.

### **Related Commands**

RECALL:SETUp, \*RCL, \*SAV

## **SAVe:WAVEform (No Query Form)**

Stores a waveform in one of the reference memory locations. This command is equivalent to selecting the Save Waveform item in the Save/Recall Waveform menu.

### **Group**

Save and Recall

### **Syntax**

SAVe:WAVEform <wfm>, REF<x>

### **Arguments**

<wfm> is CH<y> (one of the allowable channels) or MATH. This is the waveform that will be saved.

REF<x> is one of the allowable reference waveform storage locations.

### **Examples**

SAVe:WAVEFORM MATH, REFB

saves the math waveform in stored waveform memory location REFB.

### **Related Commands**

SELEct:<wfm>

## **SElect? (Query Only).**

Returns the display status of all waveforms.

### **Group**

Vertical

### **Syntax**

SElect?

### **Returns**

Waveform display status

### **Examples**

SELECT?

TDS 210 and TDS 220 (with CH2 on, Math on, and REFB on) would return :SELECT:CH1 0;CH2 1;MATH 1;REFA 0;REFB 1;

TDS 224 (with CH1 through CH4 on, Math off, and REFA and REFD on) would return :SELECT:CH1 1;CH2 1;CH3 1;CH4 1;MATH 0;REFA 1;REFB 0;REFC 0;REFD 1

### **Related Commands**

None

## **SElect:<wfm>**

Controls the display of waveforms. This command is equivalent to turning a waveform on or off from the oscilloscope front panel. To define the math waveform, use the MATH:DEFINE command (page 2-109) .

### **Group**

Vertical

### **Syntax**

SElect:<wfm> { OFF | ON | <NR1> }

SElect:<wfm>?

**Arguments**

OFF or <NR1> = 0 turns off the display of the specified waveform.

ON or <NR1>  $\neq$  0 turns on the display of the specified waveform.

<wfm> can be CH<x>, MATH, or REF<y>. See *Constructed Mnemonics* on page 2–8.

**Examples**

SELECT:CH2 ON  
displays channel 2.

SELECT:REFA?  
returns either 0 or 1, indicating whether the REFA waveform is displayed.

**Related Commands**

None

## SET? (Query Only)

Returns a string listing the oscilloscope settings except for the calibration values. You can use this string to return the oscilloscope to the state it was in when you sent SET?. This command is identical to the \*LRN? command.

**Group**

Miscellaneous

**Syntax**

SET?

---

**NOTE** The SET? query always returns a string with command headers, regardless of the setting of the HEADer command. This is because the returned string is intended to be able 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.

---

### Returns

The oscilloscope settings except for the calibration values

### Examples

SET?

a partial return string may look like this:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMAVG  
16;;HEADER 1;;VERBOSE 1;;DISPLAY:FORMAT YT;STYLE  
VECTORS;PERSISTENCE 500.0E-3;CONTRAST 50;;LOCK  
NONE;;HARDCOPY:FORMAT EPSON;PORT RS232;LAYOUT PORTRAIT;
```

### Related Commands

HEADer, \*LRN?, VERBose

## \*SRE

(Service Request Enable) sets and queries the bits in the Service Request Enable Register (SRER). For a complete discussion of the use of these registers, see page 3–1.

### Group

Status and Error

### Syntax

\*SRE <NR1>

\*SRE?

### Arguments

<NR1> is an integer value in the range from 0 to 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 its value through a power cycle.

**Examples**

\*SRE 48

sets the bits in the SRER to 00110000 binary.

\*SRE?

might return a value of 32, showing that the bits in the SRER have the binary value 00100000.

**Related Commands**

\*CLS, DESE, \*ESE, \*ESR?, EVENT?, EVMSg?, FACtory, \*PSC, \*STB?

**\*STB? (Query Only)**

(Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For a complete discussion of the use of these registers, see page 3–1.

**Group**

Status and Error

**Syntax**

\*STB?

**Returns**

<NR1>

**Examples**

\*STB?

might return the value 96, showing that the SBR contains the binary value 01100000.

**Related Commands**

\*CLS, DESE, \*ESE, \*ESR?, EVENT?, EVMSg?, FACtory, \*SRE

## **\*TRG (No Query Form)**

(Trigger) Executes commands that are defined by \*DDT.

### **Group**

Miscellaneous

### **Syntax**

\*TRG

### **Arguments**

None

### **Examples**

\*TRG

immediately executes all commands that have been defined by \*DDT.

### **Related Commands**

\*DDT

## **TRIGger**

Forces a trigger event to occur. When used as a query, TRIGger? returns the current trigger parameters.

### **Group**

Trigger

### **Syntax**

TRIGger FORCe

TRIGger?

### **Arguments**

FORCe creates a trigger event. If TRIGger:STATE is REAdy, the acquisition will complete; otherwise this command will be ignored. This is equivalent to selecting FORCE TRIGGER on the front panel.



**Examples**

TRIGGER FORCE

forces a trigger event to occur.

TRIGGER?

might return :TRIGGER:MAIN:MODE AUTO;TYPE EDGE;LEVEL  
0.0E0;HOLDOFF:VALUE 5.0E-7;;:TRIGGER:MAIN:EDGE:SOURCE  
CH1;COUPLING DC;SLOPE RISE;;:TRIGGER:MAIN:VIDEO:SOURCE  
CH1;SYNC FIELD;POLARITY NORMAL

**Related Commands**

None

**TRIGger:MAIn**

Sets the oscilloscope main trigger level to 50% of the signal's minimum and maximum value. Returns the current main trigger parameters when used as a query.

**Group**

Trigger

**Syntax**

TRIGger:MAIn SETLevel

TRIGger:MAIn?

**Arguments**

SETLevel sets the main trigger level to half way between the MIN and MAX amplitudes of the trigger source input. This is equivalent to pressing the front-panel SET LEVEL TO 50% button.

If the oscilloscope acquisition state is STOP and you send TRIGger:MAIn SETLevel, the oscilloscope ignores the command and generates event 221 (Settings conflict).

**Examples**

TRIGGER:MAIN:SETLEVEL

sets the main trigger level mid way between MAX and MIN.

**Related Commands**

None

## **TRIGger:MAIn:EDGE? (Query Only)**

Returns the trigger coupling, source, and slope settings for the main edge trigger.

**Group**

Trigger

**Syntax**

TRIGger:MAIn:EDGE?

**Returns**

Trigger coupling, source, and slope settings for the main edge trigger

**Examples**

TRIGGER:MAIN:EDGE?

might return SOURCE CH1;COUPLING DC;SLOPE RISE

**Related Commands**

None

## **TRIGger:MAIn:EDGE:COUPling**

Sets or queries the type of coupling for the main edge trigger. This is equivalent to setting Coupling in the Trigger menu.

### **Group**

Trigger

### **Syntax**

TRIGger:MAIn:EDGE:COUPling { AC | DC | HFRej | LFRej | NOISerej }

TRIGger:MAIn:EDGE:COUPling?

### **Arguments**

AC selects AC trigger coupling.

DC selects DC coupling.

HFRej coupling removes the high-frequency components of the DC signal.

LFRej coupling removes the low-frequency components of the AC signal.

NOISerej selects DC low sensitivity. It requires added signal amplitude for more stable, less false triggering.

### **Examples**

TRIGGER:MAIN:EDGE:COUPLING DC  
sets the main edge trigger coupling to DC.

### **Related Commands**

None

## TRIGger:MAIn:EDGE:SLOpe

Selects a rising or falling slope for the main edge trigger. This is equivalent to setting Slope in the Trigger menu.

### Group

Trigger

### Syntax

TRIGger:MAIn:EDGE:SLOpe { FALL | RISE }

TRIGger:MAIn:EDGE:SLOpe?

### Arguments

FALL specifies to trigger on the falling or negative edge of a signal.

RISE specifies to trigger on the rising or positive edge of a signal.

### Examples

TRIGGER:MAIN:EDGE:SLOPE RISE

sets the main edge trigger to occur on the rising slope.

### Related Commands

None

## TRIGger:MAIn:EDGE:SOURce

Sets or queries the source for the main edge trigger. This is equivalent to setting Source in the Trigger menu.

### Group

Trigger

### Syntax

(TDS 210 and TDS 220) TRIGger:MAIn:EDGE:SOURce  
{ CH<x> | EXT | EXT5 | LINE }

(TDS 224) TRIGger:MAIn:EDGE:SOURce { CH<x> | LINE }

TRIGger:MAIn:EDGE:SOURce?

**Arguments**

CH<x> specifies one of the allowable input channels.

EXT specifies the external input (TDS 210 and TDS 220).

EXT5 specifies the external input attenuated by a factor of 5 (TDS 210 and TDS 220).

LINE specifies the power line signal as a trigger source.

**Examples**

TRIGGER:MAIN:EDGE:SOURCE CH1

specifies the channel 1 as the edge trigger source.

TRIGGER:MAIN:EDGE:SOURCE?

might return CH2 for the edge trigger source.

**Related Commands**

None

**TRIGger:MAIn:HOLdoff? (Query Only)**

Returns the main trigger holdoff value.

**Group**

Trigger

**Syntax**

TRIGger:MAIn:HOLdoff?

**Returns**

Main trigger holdoff value

**Examples**

TRIGGER:MAIN:HOLDOFF?

might return :TRIGGER:MAIN:HOLDOFF:VALUE 5.0E-7.

**Related Commands**

None

## TRIGger:MAIn:HOLdoff:VALue

Sets or queries the oscilloscope main trigger holdoff value.

### Group

Trigger

### Syntax

TRIGger:MAIn:HOLdoff:VALue <NR3>

TRIGger:MAIn:HOLdoff:VALue?

### Arguments

<NR3> is the main trigger holdoff value, in the range of 500 ns to 10 s.

### Examples

TRIGGER:MAIN:HOLDOFF:VALUE 10  
sets the holdoff value to 10 s.

### Related Commands

None

## TRIGger:MAIn:LEVel

Sets the oscilloscope main trigger level. This command is equivalent to adjusting the front-panel TRIGGER LEVEL knob.

---

**NOTE** When the edge trigger source is set to AC LINE, the oscilloscope ignores the set form of the command and generates event 221 (Settings conflict).

When the edge trigger source is set to AC LINE, the query form of the command returns zero.

---

### Group

Trigger

**Syntax**

TRIGger:MAIn:LEVel <NR3>

TRIGger:MAIn:LEVel?

**Arguments**

<NR3> the main trigger level, in volts.

**Examples**

TRIGGER:MAIN:LEVEL?

might return 1.4, indicating that the main edge trigger is set to 1.4 V.

**Related Commands**

None

## TRIGger:MAIn:MODE

Sets or queries the trigger mode for the edge trigger type.

**Group**

Trigger

**Syntax**

TRIGger:MAIn:MODE { AUTO | NORMal }

TRIGger:MAIn:MODE?

**Arguments**

AUTO generates a trigger if a trigger is not detected within a specific time period. AUTO also enables scan mode for sweep speeds of 100 ms/div and slower.

NORMal waits for a valid trigger event.

**Examples**

TRIGGER:MAIN:MODE AUTO

specifies that a trigger event is automatically generated.

**Related Commands**

ACQuire:STOPAfter

## TRIGger:MAIn:TYPe

Sets or queries the type of oscilloscope trigger. This is equivalent to setting Type in the Trigger menu.

### Group

Trigger

### Syntax

TRIGger:MAIn:TYPe { EDGE | VIDEo }

TRIGger:MAIn:TYPe?

### Arguments

EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in the specified direction and is controlled by the TRIGger:MAIn:EDGE commands.

VIDEo specifies that a trigger occurs when a specified signal is found and is controlled by the TRIGger:MAIn:VIDEo commands.

### Examples

TRIGGER:MAIN:TYPE?

might return VIDEO indicating that the main trigger type is a video trigger.

### Related Commands

None

## TRIGger:MAIn:VIDEo? (Query Only)

Returns the main video trigger parameters.

### Group

Trigger

### Syntax

TRIGger:MAIn:VIDEo?



**Examples**

TRIGGER:MAIN:VIDEO?

might return :TRIGGER:MAIN:VIDEO:SOURCE CH1;SYNC FIELD;  
POLARITY NORMAL.

**Related Commands**

None

## **TRIGger:MAIn:VIDeo:POLarity**

Sets or queries the video trigger polarity. This is equivalent to selecting Polarity in the Trigger/Video menu.

**Group**

Trigger

**Syntax**

TDS 210 and TDS 220 firmware below V 2.00 with TDS2CM:  
TRIGger:MAIn:VIDeo:POLarity { INVERTed | NORMAl }

All other oscilloscope, firmware version, and module combinations:  
TRIGger:MAIn:VIDeo:POLarity { INVert | NORMAl }

TRIGger:MAIn:VIDeo:POLarity?

**Arguments**

INVERTed or INVert specifies a positive sync pulse.

NORMAl specifies a negative sync pulse.

**Examples**

TRIGGER:MAIN:VIDEO:POLARITY NORMAL  
selects negative sync polarity for the video trigger.

**Related Commands**

None

## TRIGger:MAIn:VIDeo:SOURce

Sets or queries the source for the main video trigger. This is equivalent to selecting Source in the Video trigger menu.

### Group

Trigger

### Syntax

(TRIGger:MAIn:VIDeo:SOURce { CH<x> | EXT | EXT5 }  
(TDS 210 and TDS 220)

TRIGger:MAIn:VIDeo:SOURce CH<x> (TDS 224)

TRIGger:MAIn:VIDeo:SOURce?

### Arguments

CH<x> specifies one of the allowable input channels.

EXT specifies the external input (TDS 210 and TDS 220).

EXT5 specifies the external input, attenuated by a factor of 5 (TDS 210 and TDS 220).

### Examples

TRIGGER:MAIN:VIDEO:SOURCE CH1

selects channel 1 as the source for the main video trigger.

### Related Commands

None

## TRIGger:MAIn:VIDeo:SYNc

Sets or queries the source for the video trigger sync. This is equivalent to selecting Sync in the Trigger/Video menu.

### Group

Trigger

**Syntax**

TRIGger:MAIn:VIDeo:SYNc { FIELD | LINE }

TRIGger:MAIn:VIDeo:SYNc?

**Arguments**

FIELD specifies that triggering occur on the vertical sync pulse.

LINE specifies that triggering occur on the horizontal sync pulse.

**Examples**

TRIGGER:MAIN:VIDEO:SYNc FIELD

selects the vertical sync pulse for the main video trigger sync.

**Related Commands**

None

## **TRIGger:STATE? (Query Only)**

Returns the current state of the triggering system.

Accurate real time reporting of the trigger state within any single waveform acquisition is limited by factors such as the horizontal sweep speed, and communications and task latencies.

**Group**

Trigger

**Syntax**

TRIGger:STATE?

**Returns**

ARMed indicates that the oscilloscope is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is ARMed.

REAdy indicates that all pretrigger information has been acquired and the oscilloscope is ready to accept a trigger.

TRIGger indicates that the oscilloscope has seen a trigger and is acquiring the posttrigger information.

AUTO indicates that the oscilloscope is in auto mode and acquires data even in the absence of a trigger.

SAVE indicates that acquisition is stopped or that all channels are off.

SCAN indicates that the oscilloscope is in scan mode.

**Examples**

TRIGGER:STATE?

might return READY, indicating that pretrigger data has been acquired and the oscilloscope is waiting for a trigger.

**Related Commands**

None

**\*TST? (Query Only)**

(Self-Test) Tests the GPIB interface and always returns a 0.

**Group**

Miscellaneous

**Syntax**

\*TST?

**Returns**

0

**Examples**

None

**Related Commands**

None

## UNLock (No Query Form)

Unlocks the front panel. This 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 refer to ANSI-IEEE Std. 488.1-1987, Standard Digital Interface for Programmable Instrumentation, section 2.8.3, on RL State descriptions.*

---

### Group

Miscellaneous

### Syntax

UNLock ALL

### Arguments

ALL specifies all front-panel buttons.

### Examples

UNLock ALL

unlocks all front-panel buttons and knobs so they can be used.

### Related Commands

LOCK

## VERBose

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

### **Syntax**

VERBoSe { OFF | ON | <NR1> }

VERBoSe?

### **Arguments**

ON or <NR1>  $\neq$  0 sets the Verbose state true, which returns full-length keywords for applicable setting queries.

OFF or <NR1> = 0 sets the Verbose state false, which returns minimum-length keywords for applicable setting queries.

### **Examples**

VERBoSe ON

sets the Verbose state true.

VERBoSe?

might return the value 1, showing that the Verbose state is true.

### **Related Commands**

HEADer, \*LRN?, SET?

## **\*WAI (No Query Form)**

\*WAI (wait) prevents the oscilloscope from executing further commands or queries until all pending operations finish. This command lets you synchronize the operation of the oscilloscope with your application program. Synchronization methods are described on page 3–10.

Unlike \*OPC?, \*WAI returns no value when pending operations finish. See Table 2–21 (\*OPC) on page 2–124.

### **Group**

Status and Error

### **Syntax**

\*WAI

### **Arguments**

None

**Examples**

None

**Related Commands**

BUSY?, \*OPC

**WAVFrm? (Query Only)**

Returns WFMPre? and CURVe? data for the waveform specified by the DATA:SOURce command. This command is equivalent to sending WFMPre?; CURVe?. If the waveform specified by the DATA:SOURce command is not displayed, the oscilloscope returns only the waveform transmission parameters (BYT\_Nr, BIT\_Nr, ENCdg, BN\_Fmt, BYT\_Or).

**Group**

Waveform

**Syntax**

WAVFrm?

**Returns**

See WFMPre? and CURVe? commands.

**Examples**

None

**Related Commands**

CURVe?, DATA:SOURce, WFMPre?

**WFMPre? (Query Only)**

Returns waveform transmission and formatting parameters for the waveform specified by the DATA:SOURce command.

If the waveform specified by the DATA:SOURce command is not displayed, the oscilloscope returns only the waveform transmission parameters (BYT\_Nr, BIT\_Nr, ENCdG, BN\_Fmt, BYT\_Or).

**Group**

Waveform

**Syntax**

WFMPre?

**Returns**

The format of the response (assuming the DATA:SOURce waveform is on) is:

```
BYT_Nr <NR1>;BIT_Nr <NR1>;ENCdg { ASC | BIN };
BN_Fmt { RI | RP };BYT_Or { LSB | MSB };NR_Pt <NR1>;
WFID <Qstring>;PT_FMT {ENV | Y};XINcr <NR3>;
PT Off <NR1>;XZERO <NR3>;XUnit<QString>;YMult <NR3>;
YZEro <NR3>;YOff <NR3>;YUnit <QString>
```

**Examples**

None

**Related Commands**

WAVFrm?, WFMPre:<wfm>?

## WFMPre:BIT\_Nr

Sets or queries the number of bits per waveform point for the waveform to be transferred. Changing the value of WFMPre:BIT\_Nr also changes the values of WFMPre:BYT\_Nr and DATA:WIDTH.

**Group**

Waveform

**Syntax**

WFMPre:BIT\_Nr <NR1>

WFMPre:BIT\_Nr?



**Arguments**

<NR1> is either 8 or 16, and is equivalent to `WFMPre:BYT_Nr * 8` and `DATA:WIDTH * 8`.

**Examples**

`WFMPRE:BIT_NR?`

might return 8, indicating that there are 8 bits per waveform point.

**Related Commands**

`DATA:WIDTH`, `WFMPre:BYT_Nr`

## **WFMPre:BN\_Fmt**

Sets or queries the format of binary data for the waveform to be transferred. Changing the value of `WFMPre:BN_Fmt` also changes the value of `DATA:ENCdg`.

**Group**

Waveform

**Syntax**

`WFMPre:BN_Fmt { RI | RP }`

`WFMPre:BN_Fmt?`

**Arguments**

RI specifies signed integer data-point representation.

RP specifies positive integer data-point representation.

**Examples**

`WFMPRE:BN_FMT RP`

specifies that the binary waveform data are positive integer data-points.

`WFMPRE:BN_FMT?`

returns either RI or RP as the current waveform data format.

**Related Commands**

`DATA:ENCdg`, `WFMPre:BYT_Or`, `WFMPre:ENCdg`

## WFMPre:BYT\_Nr

Sets or queries the data width for the waveform to be transferred. This command is equivalent to DATA:WIDTH. Changing WFMPre:BYT\_Nr also changes WFMPre:BIT\_Nr and DATA:WIDTH.

### Group

Waveform

### Syntax

WFMPre:BYT\_Nr <NR1>

WFMPre:BYT\_Nr?

### Arguments

<NR1> is an integer in the range of 1 to 2 that sets the number of bytes per point.

### Examples

WFMPRE:BYT\_NR 2  
specifies that there are 2 bytes per waveform data point.

### Related Commands

DATA:WIDTH, WFMPre:BIT\_Nr

## WFMPre:BYT\_Or

Sets or queries which byte of binary waveform data is transmitted first during a waveform data transfer when DATA:WIDTH or WFMPre:BYT\_Nr is set to 2, or WFMPre:BIT\_Nr is set to 16. Changing WFMPre:BYT\_Or changes DATA:ENCdg.

### Group

Waveform

### Syntax

WFMPre:BYT\_Or { LSB | MSB }

WFMPre:BYT\_Or?

**Arguments**

LSB selects the least significant byte to be transmitted first.

MSB selects the most significant byte to be transmitted first.

**Examples**

WFMPRE:BYT\_OR MSB

specifies that the most significant byte in the waveform data is transferred first.

WFMPRE:BYT\_OR?

returns either MSB or LSB depending on which data byte is transferred first.

**Related Commands**

DATA:ENCdg, WFMPre:BN\_Fmt, WFMPre:ENCdg

**WFMPre:ENCdg**

Sets or queries the type of encoding for waveform data transferred with the CURVe command. Changing WFMPre:ENCdg also changes DATA:ENCdg.

**Group**

Waveform

**Syntax**

WFMPre:ENCdg { ASC | BIN }

WFMPre:ENCdg?

**Arguments**

ASC specifies ASCII curve data.

BIN specifies binary curve data.

### Examples

WFMPRE:ENCdG ASC

specifies that the waveform data is in ASCII format.

WFMPRE:ENCdG?

might return BIN, indicating that the waveform data is in binary format.

### Related Commands

DATA:ENCdG, WFMPre:BYT\_Or, WFMPre:BN\_Fmt

## WFMPre:NR\_Pt? (Query Only)

Returns the number of points that are in the transmitted waveform record, as specified by DATA:SOURce. The number of points depends on DATA:STARt, DATA:STOP, and whether DATA:SOURce is YT or FFT. NR\_Pt is at most 2500 for YT and 1024 for FFT. NR\_Pt is always at least one.

When the DATA:SOURce is not displayed, the TDS 210 and TDS 220 (firmware below V 2.00) with a TDS2CM will return a value. All other oscilloscope, firmware version, and module combinations will generate an error and will return event code 2244.

### Group

Waveform

### Syntax

WFMPre:NR\_Pt?

### Arguments

None

### Examples

WFMPRE:NR\_Pt?

returns <NR1>, which is the number of data points. If DATA:WIDth is 2, then there are <NR1>\*2 bytes in the curve.

**Related Commands**

DAta:ENCdg, DAta:SOUrce, DAta:STARt, DAta:STOP,  
WFMPre:BN\_Fmt, WFMPre:BYT\_Or, WFMPre:<wfm>:NR\_Pt

**WFMPre:PT\_Fmt**

The set form of this command sets the format (Y or ENV) of the reference waveform as specified by the DAta:DESTination command.

The query form returns the format of the waveform specified by the DAta:SOUrce command, if that waveform is on or displayed. If the waveform is not displayed, the query form of this command generates an error and returns event code 2244.

**Group**

Waveform

**Syntax**

WFMPre:PT\_Fmt { ENV | Y }

WFMPre:PT\_Fmt?

**Arguments**

Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZEr0 + XINcr (n-PT\_Off)$$

$$Y_n = YZEr0 + YMUlt (y_n - YOFf)$$

ENV specifies that the oscilloscope transmit the waveform as minimum and maximum point pairs. Peak detect waveforms use an ENV format. Peak Detect mode specifies a maximum of 1250 (minimum, maximum) pairs, with the time between pairs being  $2 * XINcr$ . Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZEr0 + XINcr (n-PT\_Off)$$

$$Y_{n_{min}} = YZEr0 + YMUlt (y_{n_{min}} - YOFf)$$

$$Y_{n_{max}} = YZero + YMult (y_{n_{max}} - YOf)$$

### Examples

WFMPRE:PT\_FMT Y

sets the waveform data point format to Y.

WFMPRE:PT\_FMT?

returns either Y or ENV.

### Related Commands

WFMPre:<wfm>:PT\_Fmt

## WFMPre:PT\_Off

The set form of this command is ignored. The query form always returns a 0, unless the waveform is not displayed, then the query generates an error and returns event code 2244. This command is listed for compatibility with older Tektronix oscilloscopes.

### Group

Waveform

### Syntax

WFMPre:PT\_Off?

### Arguments

Arguments are ignored.

### Related Commands

HORizontal:POsition

## **WFMPre:WFId? (Query Only)**

Returns a descriptive string from the waveform specified in the DATA:SOURce command, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### **Group**

Waveform

### **Syntax**

WFMPre:WFId?

### **Arguments**

None.

### **Examples**

WFMPRE:WFID?

(YT waveform) :WFMPRE:WFID "CH1 DC COUPLING, 1.0E-1 V/DIV, 5.0E-5 S/DIV, 2500 POINTS, SAMPLE MODE"

(Non-FFT Math waveform) :WFMPRE:WFID "MATH, AC COUPLING, 5.0E0 V/DIV, 5.0E-6 S/DIV, 2500 POINTS, PK DETECT MODE"

(FFT waveform TDS2MM Only) :WFMPRE:WFID "MATH, DC COUPLING, 1.0E1 DB/DIV, 5.0E3 HZ/DIV, 1024 POINTS, SAMPLE MODE"

### **Related Commands**

WFMPre:<wfm>:WFId

## **WFMPre:XINcr**

The set form of this command specifies the interval (seconds per point for non-FFT, Hertz per point for FFT) between samples of the reference waveform specified by the DATA:DESTination command. The oscilloscope uses this value to calculate the seconds/division or Hertz/division units shown in the status bar and cursor readouts when displaying a reference waveform.

The query form returns the interval between samples of the waveform specified by the DATA:SOURce command, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### **Group**

Waveform

### **Syntax**

WFMPre:XINcr <NR3>

WFMPre:XINcr?

### **Arguments**

<NR3> is the sampling interval, in seconds or Hertz per point.

### **Examples**

None

### **Related Commands**

WFMPre:<wfm>:XINcr



## **WFMPre:XUNit (TDS2CM Version)**

The set form of this command is ignored.

The query form returns ”s”. If the waveform specified by DATA:SOURce is not displayed, the query generates event message 2244.

### **Group**

Waveform

### **Syntax**

WFMPre:XUNit

WFMPre:XUNit?

### **Examples**

None

### **Related Commands**

WFMPre:<wfm>:XUNit

## **WFMPre:XUNit (TDS2MM Version)**

The set form of this command specifies the horizontal units ("s" for seconds, "Hz" for Hertz) for the reference waveform specified by the DATA:DESTination command. Setting a reference waveform to Hz causes the oscilloscope to display the waveform as an FFT waveform.

The query form returns the horizontal units for the waveform specified by DATA:SOURce. If the waveform specified by DATA:SOURce is not displayed, the query generates event message 2244.

### **Group**

Waveform

### **Syntax**

WFMPre:XUNit <qstring>

WFMPre:XUNit?

### **Arguments**

<qstring> is "s" or "Hz".

### **Examples**

None

### **Related Commands**

WFMPre:<wfm>:XUNit

## WFMPre:XZEro

The set form of this command specifies the position, in XUNits, of the first sample of the reference waveform specified by the DATA:DESTination command, relative to the trigger.

The query form returns the position of the first sample of the waveform specified by the DATA:SOURce command, if that waveform is on or displayed.

If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

The oscilloscope sets WFMPre:XZEro to zero if the display mode was XY or if MATH FFT was the DATA:SOURce when the waveform was acquired.

---

**NOTE** *The oscilloscope uses XZEro when calculating cursor readouts.*

---

### Group

Waveform

### Syntax

WFMPre:XZEro <NR3>

WFMPre:XZEro?

### Arguments

<NR3> is the position, in XUNits, of the first sample in the waveform.

### Examples

None

### Related Commands

WFMPre:<wfm>:XZEro

## WFMPre:YMUlt

YMULT is a value, expressed in YUNits per digitizer level, used to convert waveform record values to YUNit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command sets the vertical scale factor of the reference waveform specified by the DATA:DESTination command, expressed in YUNits per digitizing level.

The query form returns a value for the waveform specified by the DATA:SOURce command, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:YMUlt <NR3>

WFMPre:YMUlt?

### Arguments

<NR3> is the vertical scale factor, in YUNits (usually volts) per sample value.

An argument value of zero has special meaning. A zero indicates that the waveform has unknown vertical scaling. An example of unknown vertical scaling is a math waveform of CH1+CH2 where CH1 vertical scaling is different from CH2 vertical scaling. In this case, WFMPre:YUNit returns U.

### Examples

None

### Related Commands

WFMPre:YUNit?

## WFMPre:YOFF

YOFF is a value, expressed in digitizer levels, used to convert waveform record values to YUNit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command stores a value for the reference waveform specified by the DATA:DESTination command. This value does not affect how the oscilloscope displays the waveform, but does affect the cursor readouts.

The query form returns a value for the waveform specified by the DATA:SOURce command, in digitizer levels, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:YOFF <NR3>

WFMPre:YOFF?

### Arguments

<NR3> is a value expressed in digitizing levels.

### Examples

None

### Related Commands

None

## WFMPre:YUNit (TDS2CM Version)

The set form of this command is ignored. Refer to the WFMPre:YMUlt command for equivalent functionality.

---

**NOTE** *The WFMPre:YMUlt command determines the vertical units of the reference waveform specified by the DATA:DESTination command. Setting YMUlt to zero specifies unknown (U) units; any other value sets the vertical scale units to V (volts).*

---

The query form returns the vertical units of the waveform specified by the DATA:SOURce command, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:YUNit?

### Returns

"Volts" for volts or "U" for units (divisions).

### Examples

None

### Related Commands

WFMPre:YUNit (TDS2MM Only), WFMPre:<wfm>:YUNit, WFMPre:YMUlt

## WFMPre:YUNit (TDS2MM Version)

The set form of this command sets the vertical units for the reference waveform specified by DATA:DESTination.

---

**NOTE** *It is possible to set a combination of WFMPre:XUNit and WFMPre:YUNit that is inconsistent (for example, seconds with dB or Hertz with volts). The oscilloscope will not warn you of this condition. The oscilloscope uses WFMPre:XUNit to determine if the waveform is a YT or an FFT.*

---

The query form returns the vertical units of the waveform specified by the DATA:SOURce command, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:YUNit <qstring>

WFMPre:YUNit?

### Arguments

<qstring> is "Volts", "U", or "dB".

### Returns

"Volts" for volts, "U" for units (divisions), or "dB" for decibels.

### Examples

None

### Related Commands

WFMPre:YUNit, WFMPre:<wfm>:YUNit, WFMPre:YMUlt

## **WFMPre:YZero (TDS 210 or TDS 220 Firmware Below V 2.00 with a TDS2CM)**

YZero is a value, expressed in YUNits, used to convert waveform record values to YUNit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command is ignored.

The query form of this command always returns "0".

### **Group**

Waveform

### **Syntax**

WFMPre:YZero?

### **Arguments**

Arguments are ignored

### **Examples**

None

### **Related Commands**

None



## **WFMPre:YZero (All Other Oscilloscope, Firmware Version, and Module Combinations)**

YZero is a value, expressed in YUNits, used to convert waveform record values to YUNit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

YZERO is used when calculating cursor readouts.

The set form of this command affects the reference waveform specified by DATA:DESTination.

The query form of this command returns a value for the waveform specified by DATA:SOUrce. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### **Group**

Waveform

### **Syntax**

WFMPre:YZero <NR3>

WFMPre:YZero?

### **Arguments**

<NR3> is a value, expressed in YUNits.

### **Examples**

None

### **Related Commands**

WFMPre:<wfm>:YZero

## WFMPre—Additional Commands for Compatibility

Table 2–22 lists additional WFMPre commands.

The set form of these commands is ignored.

The query form generates event messages 100 and 420.

These commands are included for compatibility purposes only.

**Table 2–22: Additional WFMPre commands**

Command	Argument	Description
WFMPre:XMU1t	<NR3>	Horizontal (X-axis) scale factor
WFMPre:XOfF	<NR3>	Horizontal (X-axis) offset
WFMPre:ZMU1t	<NR3>	Z-axis scale factor
WFMPre:ZOfF	<NR3>	Z-axis offset
WFMPre:ZUNit	<QString>	Z-axis units
WFMPre:ZZEro	<NR3>	Z-axis origin offset

### WFMPre:<wfm>? (Query Only)

Returns the waveform formatting data for the waveform specified by <wfm>, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates event message 420 (Query unterminated) and 2244 (Waveform requested is not on).

**Group**

Waveform

**Syntax**

WFMPre:<wfm>? (See page 2–9, *Waveform Mnemonics*)

**Returns**

The format of the response is:

```
:WFMPre:<wfm>:WFID <Qstring>;PT_FMT { ENV | Y };  
XINcr <NR3>;PT_Off <NR1>;XZEro <NR3>;XUNit <QString>;  
YMUlt <NR3>;YZEro <NR3>;YOff <NR3>;YUNit <QString>;  
NR_Pt <NR1>
```

**Examples**

None

**Related Commands**

WFMPre?

## **WFMPre:<wfm>:NR\_Pt? (Query Only) (All Oscilloscope, Firmware Version, and Module Combinations Except TDS 210 or TDS 220 Firmware Below V 2.00 with a TDS2CM)**

Returns the number of points that are in the transmitted waveform record. The number of points depends on DATA:START, DATA:STOP, and whether DATA:SOURce is YT or FFT. NR\_Pt is at most 2500 for YT and 1024 for FFT. NR\_Pt is always at least one.

When the DATA:SOURce is not displayed, the instrument will generate an error and return event code 2244.

**Group**

Waveform

**Syntax**

```
WFMPre:<wfm>:NR_Pt?
```

**Arguments**

None

**Examples**

```
WFMPRE:CH1:NR_Pt?
```

returns <NR1>, which is the number of data points. If DATA:WIDTH is 2, then there are <NR1>\*2 bytes in the curve.

### Related Commands

DATA:SOURce, DATA:STARt, DATA:STOP, WFMPre:NR\_Pt

## WFMPre:<wfm>:PT\_Fmt

The set form of this command sets the waveform format (Y or ENV) of the reference waveform specified by <wfm>, if <wfm> is a reference waveform. If <wfm> is not a reference waveform, the set command is ignored.

The query form returns the waveform format of the waveform specified by <wfm>, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:<wfm>:PT\_Fmt { ENV | Y }

WFMPre:<wfm>:PT\_Fmt?

### Arguments

Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZero + XINcr (n-PT\_Off)$$

$$Y_n = YZero + YMult (y_n - YOfF)$$

ENV specifies that the oscilloscope transmit the waveform as minimum and maximum point pairs. Peak detect waveforms use ENV format. Only y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZero + XINcr (n-PT\_Off)$$

$$Y_{n_{min}} = YZero + YMult (y_{n_{min}} - YOfF)$$

$$Y_{n_{max}} = YZero + YMult (y_{n_{max}} - YOfF)$$

**Examples**

WFMPRE:MATH:PT\_FMT?  
may return Y.

**Related Commands**

WFMPre:PT\_FMT

**WFMPre:<wfm>:PT\_Off**

The set form of this command is ignored.

The query form always returns a 0. This command is listed for compatibility with older Tektronix oscilloscopes.

**Group**

Waveform

**Syntax**

WFMPre:<wfm>:PT\_Off?

**Arguments**

Arguments are ignored.

**Examples**

WFMPRE:CH1:PT\_OFF?  
returns 0 indicating the trigger offset within the waveform record.

**Related Commands**

WFMPre:PT\_Off

**WFMPre:<wfm>:WFId? (Query Only)**

Returns a descriptive string for the waveform specified by <wfm>, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

**Group**

Waveform

### Syntax

WFMPre:<wfm>:WFId?

### Examples

WFMPRE:CH1:WFId?

(YT waveform) :WFMPRE:WFID "CH1 DC COUPLING, 1.0E-1 V/DIV, 5.0E-5 S/DIV, 2500 POINTS, SAMPLE MODE"

(Non-FFT Math waveform) :WFMPRE:WFID "MATH, AC COUPLING, 5.0E0 V/DIV, 5.0E-6 S/DIV, 2500 POINTS, PK DETECT MODE"

(FFT waveform TDS2MM Only) WFMPRE:WFID "MATH, DC COUPLING, 1.0E1 DB/DIV, 5.0E3 HZ/DIV, 1024 POINTS, SAMPLE MODE"

### Related Commands

WFMPre:WFId

## WFMPre:<wfm>:XINcr

The set form of this command specifies the interval (seconds per point for non-FFT, Hertz per point for FFT) between samples of the reference waveform specified by <wfm>, if <wfm> is a reference waveform. If <wfm> is not a reference waveform, the set command is ignored. The oscilloscope uses this value to calculate the seconds/division or Hertz/division units, shown in the status bar and cursor readouts, when displaying a reference waveform.

The query form returns the interval between samples of the waveform specified by <wfm>, if that waveform is displayed. If that waveform is not displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

---

**NOTE** For waveform records acquired in Peak Detect mode, even though there is a pair of samples for each conceptual time period, the *XINcr* is such that  $(XINcr * recl) =$  the total time over which the data was collected.

---

**Group**

Waveform

**Syntax**

WFMPre:&lt;wfm&gt;:XINcr &lt;NR3&gt;

WFMPre:&lt;wfm&gt;:XINcr?

**Arguments**

&lt;NR3&gt; is the sampling interval in XUNits.

**Examples**

WFMPre:CH1:XINcr?

might return :WFMPRE:CH1:XINCR 2.0E-6

**Related Commands**

WFMPRE:XINcr

**WFMPre:<wfm>:XUNit (TDS2CM Version)**

The set form of this command is ignored.

The query form returns ”s”. If the waveform specified by DATA:SOURce is not displayed, the query generates event message 2244.

**Group**

Waveform

**Syntax**

WFMPre:&lt;wfm&gt;:XUNit?

### Examples

WFMPRE:CH1:XUNIT?

returns "s", indicating that the horizontal unit for channel 1 is seconds.

### Related Commands

WFMPre:XUNit

## WFMPre:<wfm>:XUNit (TDS2MM Version)

The set form of this command is only valid for reference waveforms. Setting a reference waveform to Hz causes the oscilloscope to display the reference waveform as an FFT waveform.

The query form returns the horizontal units for the waveform specified by DATA:SOURce ("s" for seconds, "Hz" for Hertz). If the waveform specified by DATA:SOURce is not displayed, the query generates event message 2244.

### Group

Waveform

### Syntax

WFMPre:<wfm>:XUNit <qstring>

WFMPre:<wfm>:XUNit?

### Arguments

<qstring> is "s" or "Hz".

### Examples

None

### Related Commands

WFMPre:XUNit



## WFMPre:<wfm>:XZErO

The set form of this command specifies the position, in XUNits, relative to the trigger, of the first sample of the reference waveform specified by <wfm>. If <wfm> is not a reference waveform the set command is ignored.

The query form returns the position of the first sample of the waveform specified by the DATA:SOURce command, if that waveform is on or displayed.

If the waveform is not displayed, the query fails and the oscilloscope generates an execution error with event message 2244 ("Waveform requested is not on").

The oscilloscope sets WFMPre:<wfm>:XZErO to zero if the display mode was XY when you acquired the waveform.

---

**NOTE** *The oscilloscope uses XZErO when calculating cursor readout values.*

---

### Group

Waveform

### Syntax

WFMPre:<wfm>:XZErO <NR3>

WFMPre:<wfm>:XZErO?

### Arguments

<NR3> is the position of the first waveform sample, in XUNits.

### Returns

The format of the query return is  
:WFMPRE:<wfm>:XZErO <NR3>

### Related Commands

None

## WFMPre:<wfm>:YMUlt

The set form of this command specifies the vertical scale factor, expressed in YUNits per digitizer level, of the reference waveform specified by <wfm>. If <wfm> is not a reference waveform the set command is ignored.

The query form returns the vertical scale factor of the waveform specified by <wfm>, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:<wfm>:YMUlt <NR3>

WFMPre:<wfm>:YMUlt?

### Arguments

<NR3> is the scale factor, in YUNits (usually volts) per digitizing level.

An argument value of 0 (zero) has special meaning. A 0 indicates that the waveform has unknown vertical scaling. An example of unknown vertical scaling is a math waveform of CH1+CH2 where the vertical scaling of CH1 is different from CH2. In this case, WFMPre:<wfm>:YUNit? returns "U" (unknown).

### Examples

WFMPre:CH1:YMUlt?

might return :WFMPRE:CH1:YMULT 4.0E-2

### Related Commands

WFMPre:YMUlt, WFMPre:<wfm>:YUNit

## **WFMPre:<wfm>:YOFF**

YOFF is a value, expressed in digitizer levels, used to convert waveform record values to YUNit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command stores a value for the reference waveform specified by <wfm>. This value does not affect how the oscilloscope displays the waveform, but does affect cursor readouts. If <wfm> is not a reference waveform the set command is ignored.

The query form returns the value for the waveform specified by <wfm>, if that waveform is displayed. If that waveform is not displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### **Group**

Waveform

### **Syntax**

WFMPre:<wfm>:YOFF <NR3>

WFMPre:<wfm>:YOFF?

### **Arguments**

<NR3> is a value expressed in digitizing levels.

### **Examples**

WFMPre:CH1:YOFF?

might return :WFMPRE:CH1:YOFF 2.4E1

### **Related Commands**

WFMPre:YOFF

## WFMPre:<wfm>:YUNit (TDS2CM Version)

The set form of this command is ignored. Refer to the WFMPre:<wfm>:YMUlt command for equivalent functionality.

---

**NOTE** The WFMPre:YMUlt command determines the vertical units of the reference waveform specified by the DATA:DESTination command. Setting YMUlt to zero specifies unknown (U) units; any other value sets the vertical scale units to V (volts).

---

The query form returns the vertical units of the waveform specified by <wfm>, if that waveform is displayed. If that waveform is not displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:<wfm>:YUNit?

### Returns

<Qstring> is "V" for volts or "U" for units.

### Examples

WFMPRE:CH2:YUNIT?

might return "V", meaning that the units for the vertical component of the channel 2 waveform data are volts.

### Related Commands

WFMPre:YUNit, WFMPre:<wfm>:YMUlt

## WFMPre:<wfm>:YUNit (TDS2MM Version)

The set form of this command sets the vertical units for the reference waveform specified by DATA:DESTination.

---

**NOTE** *It is possible to set a combination of WFMPre:<wfm>:XUNit and WFMPre:<wfm>:YUNit that is inconsistent (for example, seconds with dB or Hertz with volts). The oscilloscope will not warn you of this condition. The oscilloscope uses WFMPre:<wfm>:XUNit to determine if the waveform is a YT or an FFT.*

---

The query form returns the vertical units of the waveform specified by the <wfm>, if that waveform is on or displayed. If that waveform is not on or displayed, the query fails and the oscilloscope generates an execution error with event code 2244 (waveform requested is not on).

### Group

Waveform

### Syntax

WFMPre:<wfm>:YUNit <qstring>

WFMPre:<wfm>:YUNit?

### Arguments

<qstring> is "Volts", "U", or "dB".

### Returns

"Volts" for volts, "U" for units (divisions), or "dB" for decibels. The "U" occurs when doing a CH1+CH2 math operation and the channels have different volts/div settings.

### Examples

None

### Related Commands

WFMPre:YUNit, WFMPre:<wfm>:YUNit, WFMPre:YMUlt

## **WFMPre:<wfm>YZero (TDS210 and TDS 220 Firmware Below V 2.00 with a TDS2CM)**

YZero is a value, expressed in YUNits, used to convert waveform record values to unit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command is ignored.

The query form of this command always returns "0".

### **Group**

Waveform

### **Syntax**

WFMPre:<wfm>YZero?

### **Arguments**

Arguments are ignored.

### **Examples**

WFMPre:CH1:YZero?

returns :WFMPRE:CH1:YZERO 0.0E0

### **Related Commands**

WFMPre:YZero

## **WFMPre:<wfm>:YZero (All Other Oscilloscope, Firmware Version, and Module Combinations)**

YZero is a value, expressed in YUNits, used to convert waveform record values to unit values using the following formula (where dl is digitizer levels):

$$\text{value\_in\_YUNits} = ((\text{curve\_in\_dl} - \text{YOFF\_in\_dl}) * \text{YMULT}) + \text{YZERO\_in\_YUNits}$$

The set form of this command sets a value used when calculating cursor readouts.

### **Group**

Waveform

### **Syntax**

WFMPre:<wfm>:YZero <NR3>

WFMPre:<wfm>:YZero?

### **Arguments**

<NR3> is a value, expressed in YUNits.

### **Examples**

None

### **Related Commands**

WFMPre:YZero







# Status and Events



# Status and Events

The oscilloscope provides a status and event reporting system for the GPIB and RS-232 interfaces. This system informs you of certain significant events that occur within the oscilloscope.

The oscilloscope status reporting system consists of five 8-bit registers and two queues. This section describes these registers and components, and explains how the event handling system operates.

## Registers

The registers in the event reporting system fall into two functional groups:

- The Standard Event Status Register (SESR) and the Status Byte Register (SBR) contain information about the status of the oscilloscope. These registers are the Status Registers.
- The Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER) determine whether selected types of events are reported to the Status Registers and the Event Queue. These three registers are the Enable Registers.

### Status Registers

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 service request. 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 the Enable Registers section on page 3–4.) Reading the status registers tells you what types of events have occurred.

**The Standard Event Status Register (SESR).** The SESR, shown in Figure 3–1, records eight types of events that can occur within the oscilloscope. Use \*ESR? to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events. Table 3–1 shows SESR bit functions.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

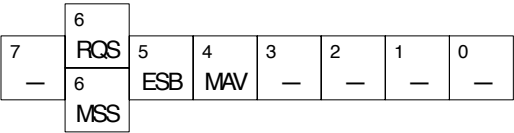
**Figure 3–1: The Standard Event Status Register (SESR)**

**Table 3–1: SESR Bit Functions**

Bit	Function
7 (MSB)	<b>PON</b> (Power On). Shows that the oscilloscope was powered on.
6	<b>URQ</b> (User Request). Not used.
5	<b>CME</b> (Command Error). Shows that an error occurred while the oscilloscope was parsing a command or query. Command error messages are listed in Table 3–4 on page 3–17.
4	<b>EXE</b> (Execution Error). Shows that an error occurred while the oscilloscope was executing a command or query. Execution error messages are listed in Table 3–5 on page 3–18.
3	<b>DDE</b> (Device Error). Shows that a device error occurred. Device error messages are listed in Table 3–6 on page 3–21.
2	<b>QYE</b> (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	<b>RQC</b> (Request Control). Not used.
0 (LSB)	<b>OPC</b> (Operation Complete). Shows that the operation is complete. This bit is set when all pending operations complete following a *OPC command.

**The Status Byte Register (SBR).** The SBR, shown in Figure 3–2, 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 (GPIB only) or \*STB? to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the \*STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits. Table 3–2 shows the SBR bit functions.



**Figure 3–2: The Status Byte Register (SBR)**

**Table 3–2: SBR Bit Functions**

Bit	Function
7 (MSB)	Not used.
6	<b>RQS</b> (Request Service), obtained from a serial poll. Shows that the oscilloscope requests service from the GPIB controller.
6	<b>MSS</b> (Master Status Summary), obtained from *STB?. Summarizes the ESB and MAV bits in the SBR.
5	<b>ESB</b> (Event Status Bit). Shows that status is enabled and present in the SESR.
4	<b>MAV</b> (Message Available). Shows that output is available in the Output Queue.
3 - 0	Not used.

**Enable Registers**

The 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 its 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.

The bits in the Enable Registers are set using various commands. The Enable Registers and the commands used to set them are described below.

**The Device Event Status Enable Register (DESER).** The DESER, shown in Figure 3–3, 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, as described earlier.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Figure 3–3: The Device Event Status Enable Register (DESER)**

**The Event Status Enable Register (ESER).** The ESER, shown in Figure 3–4, 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, and use the \*ESE? query to read it.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Figure 3–4: The Event Status Enable Register (ESER)**

**The Service Request Enable Register (SRER).** The SRER, shown in Figure 3–5, controls which bits in the SBR generate a Service RRequest (GPIB only) and are summarized by the Master Status Summary (MSS) bit.

Use the \*SRE command to set the SRER. Use the \*SRE? query to read it. The RQS bit remains set to one until either the Status Byte Register is read by a Serial Poll (GPIB only) or the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

**Figure 3–5: The Service Request Enable Register (SRER)**

### The Enable Registers and the \*PSC Command

The \*PSC command controls the contents of the Enable Registers 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 (GPIB only), 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 (GPIB only).

---

## Queues

The oscilloscope status and event reporting system contains two queues: the Output Queue and the Event Queue.

### The Output Queue

The Output Queue stores query responses waiting to be output. The oscilloscope empties the Output Queue each time it receives a new command or query message. This means you must read any query response before you send the next command or query, or you will lose responses to earlier queries. Also, an error may result.



---

**NOTE** When a controller sends a query, an <EOM>, and a second query, the digitizing 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 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.

---

### **The Event Queue**

The Event Queue stores detailed information on up to 20 events. If more than 20 events stack up in the Event Queue, the 20th event is replaced by event code 350, "Too many events."

Read the Event Queue with EVENT? (which returns only the event number), with EVMsg? (which returns the event number and a text description of the event), or with ALLEV? (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 \*ESR? to read the summary of the event from the SESR. This makes the events summarized by \*ESR? available to EVENT? and EVMSG?, and empties the SESR.

Reading the SESR erases any events that were summarized by previous \*ESR? reads but not read from the Event Queue. Events that follow an \*ESR? read are put in the Event Queue, but are not available until \*ESR? is used again.

## Event Handling Sequence

Figure 3–6 shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in Figure 3–6.

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 (GPIB only) is generated (7).

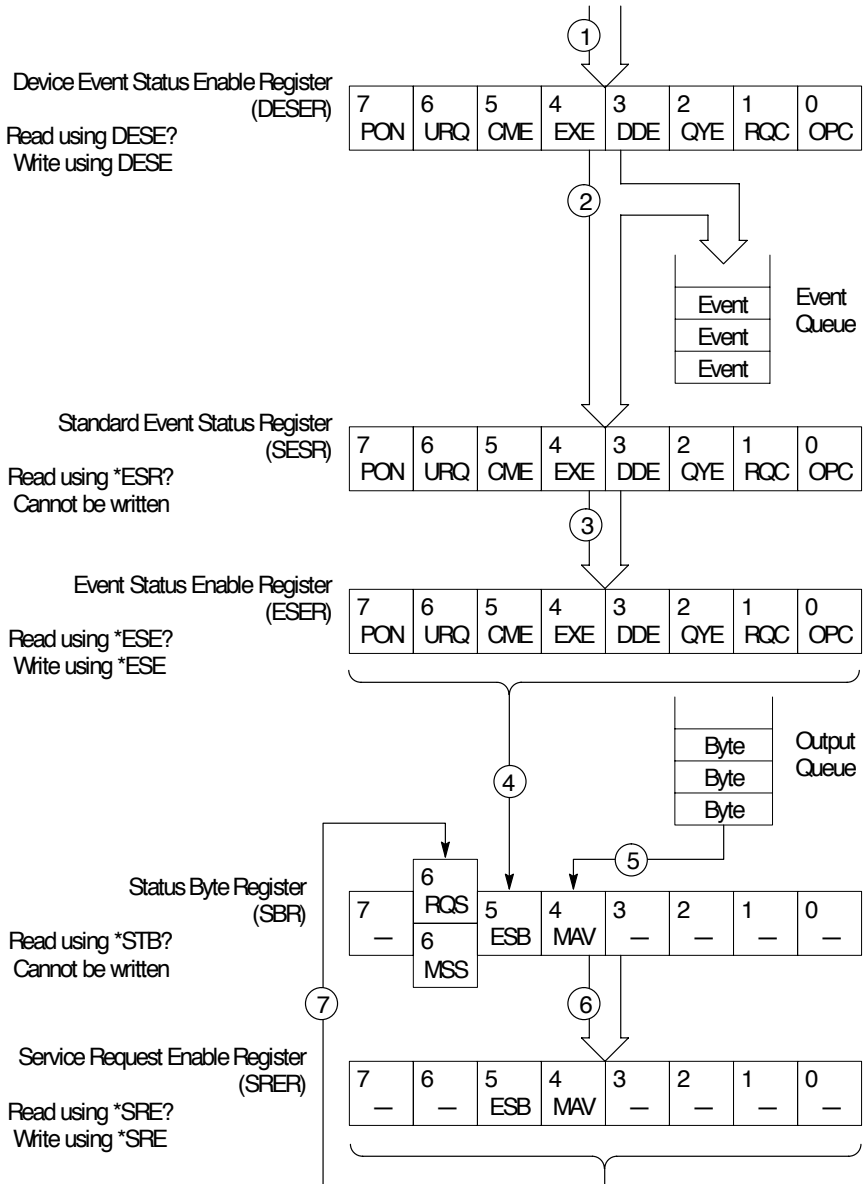


Figure 3-6: Status and Event Handling Process

## Synchronization Methods

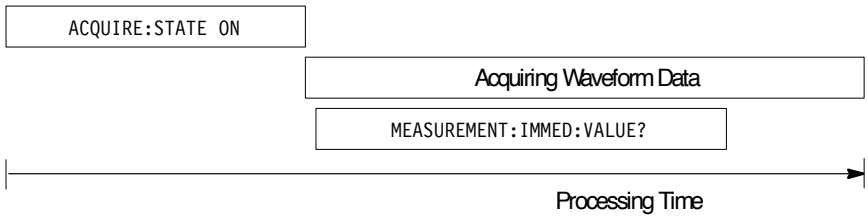
Although most commands are completed almost immediately after being received by the oscilloscope, some commands start a process that requires more time. For example, once a **HARDCOPY START** command is executed, it may be a few seconds before the hardcopy operation is complete. Rather than remain idle while the operation is in process, the oscilloscope continues processing other commands. This means that some operations are not completed in the order that they were sent.

There may be times when the result of an operation is dependent on the result of an earlier one, and you must be assured that the first operation has completed before processing the next one. The status and event reporting system provides ways to do this.

For example, a typical application would be to acquire a single-sequence waveform, and then take a measurement on the acquired waveform. You could use the following command sequence:

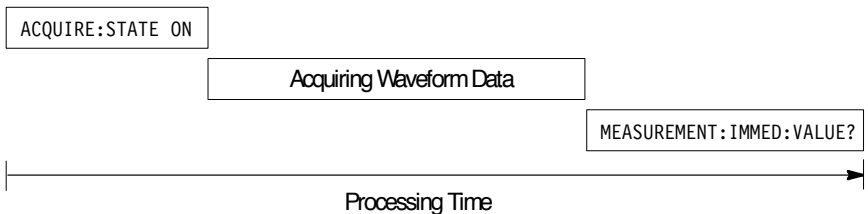
```
/** Set up single-sequence acquisition **/  
SELECT:CH1 ON  
ACQUIRE:MODE SAMPLE  
ACQUIRE:STOFAFTER SEQUENCE  
/** Acquire waveform data **/  
ACQUIRE:STATE ON  
/** Set up the measurement 2  
MEASUREMENT:IMMED:TYPE PK2PK  
MEASUREMENT:IMMED:SOURCE CH1  
/** Take peak-to-peak measurement on acquired data **/  
MEASUREMENT:IMMED:VALUE?
```

The acquisition of the waveform requires extended processing time and may not complete before the amplitude measurement is taken. (See Figure 3–7.) This will result in an incorrect peak-to-peak value.



**Figure 3-7: Command Processing Without Using Synchronization**

The acquisition of the waveform must be completed before the measurement can be taken on the acquired data. This is achieved by synchronizing the program so that the measurement command is not processed by the oscilloscope until the acquisition is complete. Figure 3-8 shows the desired processing sequence.



**Figure 3-8: Processing Sequence With Synchronization**

Four commands can be used to synchronize the operation of the oscilloscope with your application program: `*WAI`, `BUSY?`, `*OPC`, and `*OPC?`.

### Using the `*WAI` Command

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 follows:

```
/* Set up single-sequence acquisition */  
SELECT:CH1 ON  
ACQUIRE:MODE SAMPLE  
ACQUIRE:STOPAFTER SEQUENCE  
/* Acquire waveform data */  
ACQUIRE:STATE ON  
/* Set up the measurement parameters */  
MEASUREMENT:IMMED:TYPE PK2PK  
MEASUREMENT:IMMED:SOURCE CH1  
/* Wait until the acquisition is complete before taking the  
measurement */  
*WAI  
/* Take peak-to-peak measurement on acquired data */  
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, but the commands are not processed by the oscilloscope until all operations in process are complete. If the input buffer becomes full, the controller will be unable to write any more commands to the buffer and will result in a time out.

### Using the BUSY Query

BUSY? 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 BUSY? for synchronization follows:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE PK2PK
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the
measurement */
While BUSY? keep looping
/* Take peak-to-peak measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the \*WAI command. An advantage to using BUSY? is that you eliminate the possibility of a time out caused by writing too many commands to the input buffer. The controller is still tied up, though, and the repeated BUSY? results in more bus traffic.

### Using the \*OPC Command

If the corresponding status registers are enabled, the \*OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You can use this command in conjunction with either a serial poll or service request handler to achieve synchronization.

**Serial Poll Method (GPIB Only).** 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) is enabled, and the Event Status Bit (ESB) in the Status Byte Register is enabled.

The same command sequence using the \*OPC command for synchronization with serial polling looks like this:

```
/* Set up single-sequence acquisition */  
  
SELECT:CH1 ON  
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 PK2PK  
MEASUREMENT:IMMED:SOURCE CH1  
  
/* Wait until the acquisition is complete before taking the  
measurement */  
  
*OPC  
While serial poll = 0, keep looping  
  
/* Take peak-to-peak measurement on acquired data */
```



MEASUREMENT:IMMED:VALUE?

This technique requires less bus traffic than did looping on BUSY?.

**Service Request Method (GPIB Only).** 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. 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 is generated.

The same command sequence using the \*OPC command for synchronization looks like this:

```
/* Set up single-sequence acquisition */
```

```
SELECT:CH1 ON  
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 PK2PK  
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 peak-to-peak measurement on acquired data */
```

### MEASUREMENT:IMMED:VALUE?

This technique is more efficient but requires more sophisticated programming.

### Using the \*OPC Query

\*OPC? places a 1 in the Output Queue once an operation is complete. A timeout could occur if you try to read the output queue before there is any data in it.

The same command sequence using \*OPC? for synchronization follows:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE PK2PK
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the
measurement */
*OPC?
Wait for read from Output Queue.
/* Take peak-to-peak measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

Using \*OPC? synchronization is the simplest approach. It requires no status handling or loops. However, you must set the controller time out for longer than the acquisition operation.

## Messages

Tables 3–3 through 3–9 list all the programming interface messages the oscilloscope generates in response to commands and queries.

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 that follow, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

Table 3–3 shows the messages when the system has no events or status to report. These have no associated SESR bit.

**Table 3–3: No Event Messages**

Code	Message
0	No events to report : queue empty
1	No events to report : new events pending *ESR?

Table 3–4 shows the error messages generated by improper command syntax. Check that the command is properly formed and that it follows the rules in the *Command Syntax* section starting on page 2–1.

**Table 3–4: Command Error Messages - CME Bit 5**

Code	Message
100	Command error
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed

**Table 3-4: Command Error Messages - CME Bit 5 (Cont.)**

Code	Message
110	Command header error
111	Header separator error
112	Program mnemonic too long
113	Undefined header
161	Invalid block data (indefinite length blocks are not allowed over the RS-232)

Table 3-5 lists the errors that are detected during execution of a command. In these error messages, you should read “macro” as “alias.”

**Table 3-5: Execution Error Messages - EXE Bit 4**

Code	Message
200	Execution error
201	Invalid while in local
210	Trigger error
211	Trigger ignored
212	Arm ignored
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
230	Data corrupt or stale
240	Hardware error
241	Hardware missing

**Table 3–5: Execution Error Messages - EXE Bit 4 (Cont.)**

<b>Code</b>	<b>Message</b>
242	Hardware configuration error
243	Hardware I/O device error
260	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

**Table 3-5: Execution Error Messages - EXE Bit 4 (Cont.)**

<b>Code</b>	<b>Message</b>
2221	Measurement error, No valid edge: No trigger cross
2222	Measurement error, No valid edge: No second cross
2223	Measurement error, waveform mismatch
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Null Waveform
2227	Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2235	Math error, Invalid math description
2241	Waveform request is invalid
2242	Data start > record length
2243	Waveform requested is not a data source
2244	Waveform requested is not turned on
2245	Saveref error, Selected channel is turned off
2246	Saveref error, Selected channel data invalid
2248	Saveref error, Source reference data invalid
2260	Calibration error
2301	Cursor error, Off screen
2302	Cursor error, Cursors are off
2303	Cursor error, Cursor source waveform is off

Table 3–6 lists the device errors that can occur during oscilloscope operation. These errors may indicate that the oscilloscope needs repair.

**Table 3–6: Device Error Messages - DDE Bit 3**

Code	Message
300	Device-specific error
310	System error
311	Memory error
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
350	Queue overflow (does not set DDE bit)
361	Parity error in program message (check parity)
362	Framing error in program message (check baud rate)
363	Input buffer overrun (check flagging)

Table 3–7 lists the system event messages. These messages are generated whenever certain system conditions occur.

**Table 3–7: System Event Messages**

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control

**Table 3-7: System Event Messages (Cont.)**

Code	Message
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), normally caused by using a query that returns arbitrary data in a concatenated command

Table 3-8 lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

**Table 3-8: Execution Warning Messages - EXE Bit 4**

Code	Message
500	Execution warning
510	String data too long, truncated
525	Parameter underrange
526	Parameter overrange
527	Parameter rounded
528	Parameter out of range
530	Data start > stop, Values swapped internally
531	Data stop > record length, Curve truncated
532	Curve data too long, Curve truncated
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution



**Table 3–8: Execution Warning Messages - EXE Bit 4 (Cont.)**

Code	Message
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Table 3–9 lists internal errors that indicate an internal fault in the oscilloscope.

**Table 3–9: Internal Warning Messages**

Code	Message
600	Internal warning





# Appendices



# Appendix A: ASCII Code Chart

	0	1	2	3	4	5	6	7
0	<sup>0</sup> <b>NUL</b> <sub>0 0</sub>	<sup>20</sup> <b>DLE</b> <sub>10 16</sub>	<sup>40</sup> <b>SP</b> <sub>20 32</sub>	<sup>60</sup> <b>0</b> <sub>30 48</sub>	<sup>100</sup> <b>@</b> <sub>40 64</sub>	<sup>120</sup> <b>P</b> <sub>50 80</sub>	<sup>140</sup> <b>,</b> <sub>60 96</sub>	<sup>160</sup> <b>p</b> <sub>70 112</sub>
1	<sup>1</sup> <b>SOH</b> <sub>1 1</sub>	<sup>21</sup> <b>DC1</b> <sub>11 17</sub>	<sup>41</sup> <b>!</b> <sub>21 33</sub>	<sup>61</sup> <b>1</b> <sub>31 49</sub>	<sup>101</sup> <b>A</b> <sub>41 65</sub>	<sup>121</sup> <b>Q</b> <sub>51 81</sub>	<sup>141</sup> <b>a</b> <sub>61 97</sub>	<sup>161</sup> <b>q</b> <sub>71 113</sub>
2	<sup>2</sup> <b>STX</b> <sub>2 2</sub>	<sup>22</sup> <b>DC2</b> <sub>12 18</sub>	<sup>42</sup> <b>"</b> <sub>22 34</sub>	<sup>62</sup> <b>2</b> <sub>32 50</sub>	<sup>102</sup> <b>B</b> <sub>42 66</sub>	<sup>122</sup> <b>R</b> <sub>52 82</sub>	<sup>142</sup> <b>b</b> <sub>62 98</sub>	<sup>162</sup> <b>r</b> <sub>72 114</sub>
3	<sup>3</sup> <b>ETX</b> <sub>3 3</sub>	<sup>23</sup> <b>DC3</b> <sub>13 19</sub>	<sup>43</sup> <b>#</b> <sub>23 35</sub>	<sup>63</sup> <b>3</b> <sub>33 51</sub>	<sup>103</sup> <b>C</b> <sub>43 67</sub>	<sup>123</sup> <b>S</b> <sub>53 83</sub>	<sup>143</sup> <b>c</b> <sub>63 99</sub>	<sup>163</sup> <b>s</b> <sub>73 115</sub>
4	<sup>4</sup> <b>EOT</b> <sub>4 4</sub>	<sup>24</sup> <b>DC4</b> <sub>14 20</sub>	<sup>44</sup> <b>\$</b> <sub>24 36</sub>	<sup>64</sup> <b>4</b> <sub>34 52</sub>	<sup>104</sup> <b>D</b> <sub>44 68</sub>	<sup>124</sup> <b>T</b> <sub>54 84</sub>	<sup>144</sup> <b>d</b> <sub>64 100</sub>	<sup>164</sup> <b>t</b> <sub>74 116</sub>
5	<sup>5</sup> <b>ENQ</b> <sub>5 5</sub>	<sup>25</sup> <b>NAK</b> <sub>15 21</sub>	<sup>45</sup> <b>%</b> <sub>25 37</sub>	<sup>65</sup> <b>5</b> <sub>35 53</sub>	<sup>105</sup> <b>E</b> <sub>45 69</sub>	<sup>125</sup> <b>U</b> <sub>55 85</sub>	<sup>145</sup> <b>e</b> <sub>65 101</sub>	<sup>165</sup> <b>u</b> <sub>75 117</sub>
6	<sup>6</sup> <b>ACK</b> <sub>6 6</sub>	<sup>26</sup> <b>SYN</b> <sub>16 22</sub>	<sup>46</sup> <b>&amp;</b> <sub>26 38</sub>	<sup>66</sup> <b>6</b> <sub>36 54</sub>	<sup>106</sup> <b>F</b> <sub>46 70</sub>	<sup>126</sup> <b>V</b> <sub>56 86</sub>	<sup>146</sup> <b>f</b> <sub>66 102</sub>	<sup>166</sup> <b>v</b> <sub>76 118</sub>
7	<sup>7</sup> <b>BEL</b> <sub>7 7</sub>	<sup>27</sup> <b>ETB</b> <sub>17 23</sub>	<sup>47</sup> <b>'</b> <sub>27 39</sub>	<sup>67</sup> <b>7</b> <sub>37 55</sub>	<sup>107</sup> <b>G</b> <sub>47 71</sub>	<sup>127</sup> <b>W</b> <sub>57 87</sub>	<sup>147</sup> <b>g</b> <sub>67 103</sub>	<sup>167</sup> <b>w</b> <sub>77 119</sub>
8	<sup>10</sup> <b>BS</b> <sub>8 8</sub>	<sup>30</sup> <b>CAN</b> <sub>18 24</sub>	<sup>50</sup> <b>(</b> <sub>28 40</sub>	<sup>70</sup> <b>8</b> <sub>38 56</sub>	<sup>110</sup> <b>H</b> <sub>48 72</sub>	<sup>130</sup> <b>X</b> <sub>58 88</sub>	<sup>150</sup> <b>h</b> <sub>68 104</sub>	<sup>170</sup> <b>x</b> <sub>78 120</sub>
9	<sup>11</sup> <b>HT</b> <sub>9 9</sub>	<sup>31</sup> <b>EM</b> <sub>19 25</sub>	<sup>51</sup> <b>)</b> <sub>29 41</sub>	<sup>71</sup> <b>9</b> <sub>39 57</sub>	<sup>111</sup> <b>I</b> <sub>49 73</sub>	<sup>131</sup> <b>Y</b> <sub>59 89</sub>	<sup>151</sup> <b>i</b> <sub>69 105</sub>	<sup>171</sup> <b>y</b> <sub>79 121</sub>

*Continued on next page*

Appendix A: ASCII Code Chart

	0	1	2	3	4	5	6	7
A	<sup>12</sup> LF <sub>A 10</sub>	<sup>32</sup> SUB <sub>1A 26</sub>	<sup>52</sup> * <sub>2A 42</sub>	<sup>72</sup> : <sub>3A 58</sub>	<sup>112</sup> J <sub>4A 74</sub>	<sup>132</sup> Z <sub>5A 90</sub>	<sup>152</sup> j <sub>6A 106</sub>	<sup>172</sup> z <sub>7A 122</sub>
B	<sup>13</sup> VT <sub>B 11</sub>	<sup>33</sup> ESC <sub>1B 27</sub>	<sup>53</sup> + <sub>2B 43</sub>	<sup>73</sup> ; <sub>3B 59</sub>	<sup>113</sup> K <sub>4B 75</sub>	<sup>133</sup> [ <sub>5B 91</sub>	<sup>153</sup> k <sub>6B 107</sub>	<sup>173</sup> { <sub>7B 123</sub>
C	<sup>14</sup> FF <sub>C 12</sub>	<sup>34</sup> FS <sub>1C 28</sub>	<sup>54</sup> , <sub>2C 44</sub>	<sup>74</sup> < <sub>3C 60</sub>	<sup>114</sup> L <sub>4C 76</sub>	<sup>134</sup> \ <sub>5C 92</sub>	<sup>154</sup> l <sub>6C 108</sub>	<sup>174</sup>   <sub>7C 124</sub>
D	<sup>15</sup> CR <sub>D 13</sub>	<sup>35</sup> GS <sub>1D 29</sub>	<sup>55</sup> - <sub>2D 45</sub>	<sup>75</sup> = <sub>3D 61</sub>	<sup>115</sup> M <sub>4D 77</sub>	<sup>135</sup> ] <sub>5D 93</sub>	<sup>155</sup> m <sub>6D 109</sub>	<sup>175</sup> } <sub>7D 125</sub>
E	<sup>16</sup> SO <sub>E 14</sub>	<sup>36</sup> RS <sub>1E 30</sub>	<sup>56</sup> . <sub>2E 46</sub>	<sup>76</sup> > <sub>3E 62</sub>	<sup>116</sup> N <sub>4E 78</sub>	<sup>136</sup> ^ <sub>5E 94</sub>	<sup>156</sup> n <sub>6E 110</sub>	<sup>176</sup> ~ <sub>7E 126</sub>
F	<sup>17</sup> SI <sub>F 15</sub>	<sup>37</sup> US <sub>1F 31</sub>	<sup>57</sup> / <sub>2F 47</sub>	<sup>77</sup> ? <sub>3F 63</sub>	<sup>117</sup> O <sub>4F 79</sub>	<sup>137</sup> _ <sub>5F 95</sub>	<sup>157</sup> o <sub>6F 111</sub>	<sup>177</sup> DEL (RUBOUT) <sub>7F 127</sub>

KEY

octal

25

hex

15

21

decimal

NAK

ASCII character

## Appendix B: Factory Setup

The following listing is the instrument response to the concatenated command **FACTory;SET?**. This response describes the factory default setup in detail. (Carriage returns have been inserted for clarity.)

Items enclosed in **< >** brackets are return values associated with the **TDS2MM** module.

Items enclosed in **( )** parentheses are returned by the **SET?** query response, but are not changed by the **FACTory** command.

### **TDS 210 and TDS 220**

```
:HEADER 1;(:VERBOSE 1;)

:DATA:ENCDG RIBINARY;DESTINATION REFA;SOURCE CH1;
      START 1;STOP 2500;WIDTH 1;

(:LOCK NONE;)

:DISPLAY:FORMAT YT;STYLE VECTORS;PERSISTENCE 0;
      CONTRAST 50;

:ACQUIRE:MODE SAMPLE;NUMAVG 16;STATE 1;
      STOPAFTER RUNSTOP;

:CH1:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
      COUPLING DC;BANDWIDTH OFF;<INVERT OFF;>

:CH2:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
      COUPLING DC;BANDWIDTH OFF;<INVERT OFF;>

:HORIZONTAL:VIEW MAIN;
      MAIN:SCALE 5.0E-4;POSITION 0.0E0;
      :HORIZONTAL:DELAY:SCALE 5.0E-5;POSITION 0.0E0;

:TRIGGER:MAIN:MODE AUTO;TYPE EDGE;
      HOLDOFF:VALUE 5.0E-7;
      :TRIGGER:MAIN:EDGE:SOURCE CH1;COUPLING DC;
      SLOPE RISE;:TRIGGER:MAIN:VIDEO:SOURCE CH1;
      SYNC LINE;POLARITY NORMAL;
      :TRIGGER:MAIN:LEVEL 0.0E0;
```

```
:SELECT:CH1 1;CH2 0;MATH 0;REFA 0;REFB 0;
:CURSOR:FUNCTION OFF;SELECT:SOURCE CH1;
:CURSOR:VBARS:UNITS SECONDS;
POSITION1 -2.0E-3; POSITION2 2.0E-3;
:CURSOR:HBARS:POSITION1 -3.2E0; POSITION2 3.2E0;
:MEASUREMENT:MEAS1:TYPE NONE;SOURCE CH1;
:MEASUREMENT:MEAS2:TYPE NONE;SOURCE CH1;
:MEASUREMENT:MEAS3:TYPE NONE;SOURCE CH1;
:MEASUREMENT:MEAS4:TYPE NONE;SOURCE CH1;
:MEASUREMENT:IMMED:TYPE PERIOD;SOURCE CH1;
:MATH:DEFINE "CH1 + CH2";
<FFT:HORIZONTAL:POSITION 5.0E1;SCALE 1.0E0;>
<:MATH:FFT:VERTICAL:POSITION 0.0E0;SCALE 1.0E0;>
(:HARDCOPY:FORMAT EPSON;PORT CENTRONICS;
LAYOUT PORTRAIT;)
(:LANGUAGE ENGLISH)
```

## **TDS 224**

```
:HEADER 1;(:VERBOSE 1;)
:DATA:ENDCG RIBINARY;DESTINATION REFA;SOURCE CH1;
START 1;STOP 2500;WIDTH 1;
(:LOCK NONE;)
:DISPLAY:FORMAT YT;STYLE VECTORS;PERSISTENCE 0;
CONTRAST 50;
:ACQUIRE:MODE SAMPLE;NUMAVG 16;STATE 1;
STOPAFTER RUNSTOP;
:CH1:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
COUPLING DC;BANDWIDTH OFF;INVERT OFF;
:CH2:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
COUPLING DC;BANDWIDTH OFF;INVERT OFF;
:CH3:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
COUPLING DC;BANDWIDTH OFF;INVERT OFF;
:CH4:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;
COUPLING DC;BANDWIDTH OFF;INVERT OFF;
```



```
:HORIZONTAL:VIEW MAIN;MAIN:SCALE 5.0E-4;  
  POSITION 0.0E0;:HORIZONTAL:DELAY:SCALE 5.0E-5;  
  POSITION 0.0E0;  
  
:TRIGGER:MAIN:MODE AUTO;TYPE EDGE;  
  HOLDOFF:VALUE 5.0E-7;  
  :TRIGGER:MAIN:EDGE:SOURCE CH1;COUPLING DC;  
  SLOPE RISE;:TRIGGER:MAIN:VIDEO:SOURCE CH1;  
  SYNC LINE;POLARITY NORMAL;  
  :TRIGGER:MAIN:LEVEL 0.0E0;  
  
:SELECT:CH1 1;CH2 0;CH3 0;CH4 0;MATH 0;REFA 0;  
  REFB 0;REFC 0;REFD 0;  
  
:CURSOR:FUNCTION OFF;SELECT:SOURCE CH1;  
  
:CURSOR:VBARS:UNITS SECONDS;POSITION1 -2.0E-3;  
  POSITION2 2.0E-3;  
  :CURSOR:HBARS:POSITION1 -3.2E0;POSITION2 3.2E0;  
  
:MEASUREMENT:MEAS1:TYPE NONE;SOURCE CH1;  
  :MEASUREMENT:MEAS2:TYPE NONE;SOURCE CH1;  
  :MEASUREMENT:MEAS3:TYPE NONE;SOURCE CH1;  
  :MEASUREMENT:MEAS4:TYPE NONE;SOURCE CH1;  
  :MEASUREMENT:IMMED:TYPE PERIOD;SOURCE CH1;  
  
:MATH:DEFINE "CH1 - CH2";  
  <FFT:HORIZONTAL:POSITION 5.0E1;SCALE 1.0E0;>  
  <:MATH:FFT:VERTICAL:POSITION 0.0E0;SCALE 1.0E0;>  
  
(:HARDCOPY:FORMAT EPSON;PORT CENTRONICS;  
  LAYOUT PORTRAIT;)  
  
(:LANGUAGE ENGLISH)
```





# Glossary and Index



# Glossary

## **ASCII**

Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the digitizing oscilloscope using ASCII character encoding.

## **Address**

A 7-bit code that identifies an instrument on the communication bus. The digitizing oscilloscope must have a unique address for the controller to recognize and transmit commands to it.

## **Backus-Naur Form (BNF)**

A standard notation system for command syntax. The syntax in this manual use BNF notation.

## **Controller**

A computer or other device that sends commands to and accepts responses from the digitizing oscilloscope.

## **EOI**

A mnemonic referring to the control line “End or Identify” on the GPIB interface bus. One of the two possible end-of-message terminators.

## **EOM**

A generic acronym referring to the end-of-message terminator. The end-of-message terminator is either an EOI or the ASCII code for line feed (LF).

## **IEEE**

Acronym for the Institute of Electrical and Electronics Engineers.

## **QuickBASIC**

A computer language (distributed by Microsoft) that is based on the Beginner’s All-Purpose Symbolic Instruction Code.

**QuickC**

A computer language (distributed by Microsoft) that is based on the C programming language.

**RS-232**

A serial, full-duplex, asynchronous communication port that follows ANSI/EIA/TIA-562-1989[1], ANSI/EIA/TIA-574-1990[2], and CCITT V.24-1989[3] standards.

# Index

## Symbols

\*OPC, 3–14

\*WAI, 3–11

## A

Abbreviating, commands, 2–5

ACQUIRE?, 2–33

ACQUIRE:MODE, 2–34

ACQUIRE:NUMACQ?, 2–35

ACQUIRE:NUMAVG, 2–36

ACQUIRE:STATE, 2–37

ACQUIRE:STOPAFTER, 2–38

Acquisition command group, 2–15

Acquisition commands

ACQUIRE?, 2–33

ACQUIRE:MODE, 2–34

ACQUIRE:NUMACQ?, 2–35

ACQUIRE:NUMAVG, 2–36

ACQUIRE:STATE, 2–37

ACQUIRE:STOPAFTER, 2–38

ALLEV?, 2–39

Argument, command, 2–2

ASCII, 2–1

AUTOSET, 2–40

## B

Block, command argument, 2–12

BNF (Backus-Naur form), 2–1

Break signal, 2–5

BUSY?, 2–40, 3–13

## C

\*CAL?, 2–42

CALibrate:ABORT, 2–43

CALibrate:CONTINUE, 2–44

CALibrate:FACTory, 2–44

CALibrate:INTERNAL, 2–45

CALibrate:STATUS?, 2–46

Calibration and diagnostic command group, 2–16

Calibration and diagnostic commands

\*CAL?, 2–42

CALibrate:ABORT, 2–43

CALibrate:CONTINUE, 2–44

CALibrate:FACTory, 2–44

CALibrate:INTERNAL, 2–45

CALibrate:STATUS?, 2–46

DIAG:RESULT:FLAG?, 2–75

DIAG:RESULT:LOG?, 2–76

ERRORLOG:FIRST?, 2–81

ERRORLOG:NEXT?, 2–81

CH<x>?, 2–46

CH<x>:BANDWIDTH, 2–47

CH<x>:COUPLING, 2–48

CH<x>:INVERT, 2–49

CH<x>:POSITION, 2–49

CH<x>:PROBE, 2–51

CH<x>:SCALE, 2–52

CH<x>:VOLTS, 2–53

Channel, command mnemonic, 2–8

CH<x>, command mnemonic, 2–8

Clear status, 2–54

\*CLS, 2–54

Command

\*OPC, 3–14

\*WAI, 3–11

abbreviating, 2–5

argument, 2–2

block argument, 2–12

BUSY?, 3–13

- common, 2-21, 2-23
- concatenating, 2-6
- header, 2-2
- message, 2-2
- mnemonic, 2-2
- query, 2-2
- rules for forming, 2-1
- separator, 2-2
- set, 2-2
- synchronizing, 3-10
- syntax, 2-1
  - BNF (Backus-Naur form), 2-1
- Command argument
  - numeric, 2-9
  - quoted string, 2-10
- Command entry rules, 2-5
- Command group
  - acquisition, 2-15
  - calibration and diagnostic, 2-16
  - cursor, 2-17
  - display, 2-18
  - hardcopy, 2-18
  - horizontal, 2-19
  - measurement, 2-20
  - miscellaneous, 2-21
  - RS-232, 2-22
  - save and recall, 2-23
  - status and error, 2-23
  - trigger, 2-24
  - vertical, 2-25
  - waveform, 2-26
- Command mnemonic
  - channel, 2-8
  - CH<x>, 2-8
  - measurement specifier, 2-9
  - MEAS<x>, 2-9
  - reference waveform, 2-8
  - REF<x>, 2-8
  - waveform, 2-9
  - <wfm>, 2-9
- Command syntax, 2-1
  - BNF (Backus-Naur form), 2-1
- Common command, 2-23
- Common GPIB commands
  - \*CAL?, 2-42
  - CALibrate:CONTINUE, 2-44
  - CALibrate:FACTory, 2-44
  - CALibrate:INTERNAL, 2-45
  - CALibrate:STATUS?, 2-46
  - \*CLS, 2-54
- Concatenating, command, 2-6
- Configuration, command query, 2-105
- Cursor command group, 2-17
- Cursor commands
  - CURSOR?, 2-55
  - CURSOR:FUNCTION, 2-55
  - CURSOR:HBARS?, 2-56
  - CURSOR:HBARS:DELTA?, 2-57
  - CURSOR:HBARS:POSITION<x>, 2-58
  - CURSOR:HBARS:UNITS?, 2-59
  - CURSOR:SELECT:SOURCE, 2-59
  - CURSOR:VBARS?, 2-60
  - CURSOR:VBARS:DELTA?, 2-61
  - CURSOR:VBARS:POSITION<x>, 2-62
  - CURSOR:VBARS:UNITS, 2-63
- CURSOR?, 2-55
- CURSOR:FUNCTION, 2-55
- CURSOR:HBARS?, 2-56
- CURSOR:HBARS:DELTA?, 2-57
- CURSOR:HBARS:POSITION<x>, 2-58
- CURSOR:HBARS:UNITS?, 2-59



CURSOR:SELECT:SOURCE,  
2–59  
CURSOR:VBARS?, 2–60  
CURSOR:VBARS:DELTA?, 2–61  
CURSOR:VBARS:POSITION<x>,  
2–62  
CURSOR:VBARS:UNITS, 2–63  
CURVE, 2–64

## D

DATA, 2–65  
DATA:DESTINATION, 2–66  
DATA:ENCDG, 2–67  
DATA:SOURCE, 2–69  
DATA:START, 2–70  
DATA:STOP, 2–71  
DATA:TARGET, 2–72  
DATA:WIDTH, 2–73  
DCL, 2–5, 3–7  
\*DDT, 2–73  
DESE command, 2–74, 3–4  
DESER register, 2–74, 2–125, 3–4  
Device clear, 2–5  
DIAG:RESULT:FLAG?, 2–75  
DIAG:RESULT:LOG?, 2–76  
Display command group, 2–18  
Display commands  
  DISPLAY?, 2–77  
  DISPLAY:CONTRAST, 2–77  
  DISPLAY:FORMAT, 2–78  
  DISPLAY:PERSISTENCE, 2–79  
  DISPLAY:STYLE, 2–80  
DISPLAY?, 2–77  
DISPLAY:CONTRAST, 2–77  
DISPLAY:FORMAT, 2–78  
DISPLAY:PERSISTENCE, 2–79  
DISPLAY:STYLE, 2–80

## E

Entering commands, 2–5  
Entering incorrect numeric arguments, 2–10  
EOM (end of message), 2–7  
Error message, programming interface, 3–17  
ERRORLOG:FIRST?, 2–81  
ERRORLOG:NEXT?, 2–81  
\*ESE, 2–82, 3–5  
ESER register, 2–82, 2–125, 3–5  
\*ESR?, 2–83  
\*ESR? query, 3–2  
Event handling, 3–1  
Event query, 2–84  
Event queue, 2–84, 3–7  
Event queue sequence, 3–8  
EVENT?, 2–84  
EVMSG?, 2–84  
EVQTY?, 2–85  
Examples  
  block argument, 2–12  
  quoted strings, 2–11

## F

FACTORY, 2–86  
Factory setup, detailed description,  
  B–1

## G

GPIO, EOM (end of message), 2–7

# H

## Hard copy commands

- HARDCOPY, 2–88
- HARDCOPY:FORMAT, 2–89
- HARDCOPY:LAYOUT, 2–90
- HARDCOPY:PORT, 2–91

## HARDCOPY, 2–88

## Hardcopy command group, 2–18

- HARDCOPY:FORMAT, 2–89
- HARDCOPY:LAYOUT, 2–90
- HARDCOPY:PORT, 2–91

## HDR, 2–91

## HEADER, 2–92

## Header

- command, 2–2, 2–92
- included in query response, 2–92, 2–155

## Horizontal command group, 2–19

## Horizontal commands

- HORIZONTAL?, 2–93
- HORIZONTAL:DELAY?, 2–94
- HORIZONTAL:DELAY:POSITION, 2–94, 2–97, 2–100
- HORIZONTAL:DELAY:SCALE, 2–95
- HORIZONTAL:DELAY:SECDIV, 2–96
- HORIZONTAL:MAIN?, 2–97
- HORIZONTAL:MAIN:SCALE, 2–98
- HORIZONTAL:MAIN:SECDIV, 2–99
- HORIZONTAL:RECORD-LENGTH, 2–101
- HORIZONTAL:SCALE, 2–101
- HORIZONTAL:SECDIV, 2–102
- HORIZONTAL:TRIGGER:POSITION, 2–103
- MATH:FFT:HORIZONTAL:POSITION, 2–111

- MATH:FFT:HORIZONTAL:SCALE, 2–112

## HORIZONTAL?, 2–93

## HORIZONTAL:DELAY?, 2–94

## HORIZONTAL:DELAY:POSITION, 2–94, 2–97, 2–100

## HORIZONTAL:DELAY:SCALE, 2–95

## HORIZONTAL:DELAY:SECDIV, 2–96

## HORIZONTAL:MAIN?, 2–97

## HORIZONTAL:MAIN:SCALE, 2–98

## HORIZONTAL:MAIN:SECDIV, 2–99

## HORIZONTAL:RECORD-LENGTH, 2–101

## HORIZONTAL:SCALE, 2–101

## HORIZONTAL:SECDIV, 2–102

## HORIZONTAL:TRIGGER:POSITION, 2–103

# I

## ID?, 2–104

## \*IDN?, 2–105

## IEEE Std. 488.2–1987, 2–1, 2–21, 2–23

# L

## LANGUAGE, 2–106

## LOCK, 2–107

## \*LRN?, 2–108

# M

## Manual trigger, simulation with command, 2–142

- 
- MATH?, 2–109
  - MATH:DEFINE, 2–109
  - MATH:FFT:HORIZONTAL:POSITION, 2–111
  - MATH:FFT:HORIZONTAL:SCALE, 2–112
  - MATH:FFT:VERTICAL:POSITION, 2–113
  - MATH:FFT:VERTICAL:SCALE, 2–113
  - Measurement command group, 2–20
  - Measurement commands
    - MEASUREMENT?, 2–114
    - MEASUREMENT:IMMED?, 2–115
    - MEASUREMENT:IMMED:SOURCE1, 2–115
    - MEASUREMENT:IMMED:TYPE, 2–117
    - MEASUREMENT:IMMED:UNITS?, 2–118
    - MEASUREMENT:IMMED:VALUE?, 2–119
    - MEASUREMENT:MEAS<x>?, 2–120
    - MEASUREMENT:MEAS<x>:TYPE, 2–121
    - MEASUREMENT:MEAS<x>:UNITS?, 2–122
    - MEASUREMENT:MEAS<x>:VALUE?, 2–123
    - MEASUREMENT:MEAS<x>:SOURCE, 2–120
  - Measurement specifier, command mnemonic, 2–9
  - MEASUREMENT?, 2–114
  - MEASUREMENT:IMMED?, 2–115
  - MEASUREMENT:IMMED:SOURCE1, 2–115
  - MEASUREMENT:IMMED:TYPE, 2–117
  - MEASUREMENT:IMMED:UNITS?, 2–118
  - MEASUREMENT:IMMED:VALUE?, 2–119
  - MEASUREMENT:MEAS<x>?, 2–120
  - MEASUREMENT:MEAS<x>:TYPE, 2–121
  - MEASUREMENT:MEAS<x>:UNITS?, 2–122
  - MEASUREMENT:MEAS<x>:VALUE?, 2–123
  - MEASUREMENT:MEAS<x>:SOURCE, 2–120
  - MEAS<x>, command mnemonic, 2–9
  - Message
    - command, 2–2
    - command terminator, 2–7
    - handling, 3–1
    - table of program messages, 3–17
  - Miscellaneous command group, 2–21
  - Miscellaneous commands
    - AUTOSET, 2–40
    - \*DDT, 2–73
    - FACTORY, 2–86
    - HDR, 2–91
    - HEADER, 2–92
    - \*IDN?, 2–105
    - LANGUAGE, 2–106
    - LOCK, 2–107
    - \*LRN?, 2–108

- REM, 2–128
- SET, 2–139
- \*TRG, 2–142
- UNLOCK, 2–155
- VERBOSE, 2–155
- Mnemonic, command, 2–2

## N

- Numeric, command argument, 2–9
- Numeric argument coercion, 2–10

## O

- \*OPC, 2–124
- Operation complete command, 2–124
- Operation complete wait, 2–156
- Output queue, 3–6

## P

- Power-on status clear command, 2–125
- Preamble, waveform, 2–31
- \*PSC, 2–125
- \*PSC command, 3–6

## Q

- Query, header in query response, 2–92, 2–155
- Query command, 2–2
- Queue
  - event, 3–7
  - event handling sequence, 3–8
  - output, 3–6
- Quoted string, command argument, 2–10

- Quoted string examples, 2–11
- Quoted string rules, 2–11

## R

- \*RCL, 2–126
- Recall setting command, 2–126
- RECALL:SETUP, 2–127
- Reference waveform, command mnemonic, 2–8
- REF<x>, command mnemonic, 2–8
- Register
  - DESER, 2–74, 2–125, 3–4
  - ESER, 2–82, 2–125, 3–5
  - SBR, 2–141, 3–3
  - SESR, 2–54, 2–83, 2–124, 3–2
  - SRER, 2–125, 2–140, 3–5
- REM, 2–128
- Reset
  - command, 2–134
  - to factory settings, 2–86
- RS-232 command group, 2–22
- RS232 commands
  - RS232?, 2–128
  - RS232:BAUD, 2–129
  - RS232:HARDFLAGGING, 2–130
  - RS232:PARITY, 2–131
  - RS232:SOFTFLAGGING, 2–132
  - RS232:TRANSMIT:TERMINATOR, 2–133
- RS232?, 2–128
- RS232:BAUD, 2–129
- RS232:HARDFLAGGING, 2–130
- RS232:PARITY, 2–131
- RS232:SOFTFLAGGING, 2–132
- RS232:TRANSMIT:TERMINATOR, 2–133
- \*RST, 2–134

Rules, command forming, 2-1

Rules, command entry, 2-5

Rules, quoted strings, 2-11

## S

\*SAV, 2-135

Save and recall command group,  
2-23

Save and recall commands

\*RCL, 2-126

RECALL:SETUP, 2-127

\*SAV, 2-135

SAVE:SETUP, 2-136

SAVE:WAVEFORM, 2-137

Save setting command, 2-135

SAVE:SETUP, 2-136

SAVE:WAVEFORM, 2-137

SBR register, 2-141, 3-3

SELECT?, 2-138

SELECT:<wfm>, 2-138

Self test, 2-154

Separator, command, 2-2

Serial poll, 3-3

Service request enable command,  
2-140

Service request enable register,  
2-140

SESR register, 2-54, 2-83, 2-124,  
3-2

Set command, 2-2

SET?, 2-139

Setting

command query, 2-108

query, 2-108

recall command, 2-126

save command, 2-135

Setups, factory setup description,  
B-1

\*SRE command, 2-140, 3-5

SRER register, 2-125, 2-140, 3-5

Status, 3-1

Status and error command group,  
2-23

Status and error commands

ALLEV?, 2-39

BUSY?, 2-40

\*CLS, 2-54

DESE, 2-74, 3-4

\*ESE, 2-82, 3-5

\*ESR?, 2-83, 3-2

EVENT?, 2-84

EVMSG?, 2-84

EVQTY?, 2-85

ID?, 2-104

\*OPC, 2-124

\*PSC, 2-125, 3-6

\*RST, 2-134

\*SRE, 2-140, 3-5

\*STB?, 2-141, 3-3

\*TST?, 2-154

\*WAI, 2-156

\*STB?, 2-141

\*STB? query, 3-3

Synchronizing commands, 3-10

Syntax

BNF (Backus-Naur form), 2-1  
command, 2-1

## T

Table, programming message, 3-17

TDS2MM-related commands

CH<x>:INVERT, 2-49

MATH:DEFINE, 2-109

MATH:FFT:HORIZON-  
TAL:POSITION, 2-111

MATH:FFT:HORIZON-  
TAL:SCALE, 2-112

- MATH:FFT:VERTICAL:POSITION, 2–113
- MATH:FFT:VERTICAL:SCALE, 2–113
- MEASURE-  
MENT:IMMED:TYPE, 2–117
- MEASURE-  
MENT:MEAS<X>:TYPE, 2–121
- WFMPRE:<wfm>:WFID?, 2–179
- WFMPRE:<wfm>:XINCR, 2–180
- WFMPRE:<wfm>:XUNIT, 2–182
- WFMPRE:<wfm>:XZERO, 2–183
- WFMPRE:<wfm>:YMULT, 2–184
- WFMPRE:<wfm>:YOFF, 2–185
- WFMPRE:<wfm>:YUNIT, 2–187
- WFMPRE:WFID?, 2–165
- WFMPRE:XINCR, 2–166, 2–168
- WFMPRE:XZERO, 2–169
- WFMPRE:YMULT, 2–170
- WFMPRE:YOFF, 2–171
- WFMPRE:YUNIT, 2–173
- WFMPRE:YZERO, 2–175, 2–189
- Tek Std. Codes and Formats 1989, 2–21, 2–23
- Terminator, command message, 2–7
- Time base, manual trigger simulation, 2–142
- Transferring data
  - from the oscilloscope, 2–32
  - to the oscilloscope, 2–32
- \*TRG, 2–142
- TRIGGER, 2–142
- Trigger command group, 2–24
- Trigger commands
  - TRIGGER, 2–142
  - TRIGGER:MAIN, 2–143
  - TRIGGER:MAIN:EDGE?, 2–144
  - TRIGGER:MAIN:EDGE:COUPLING, 2–145
  - TRIGGER:MAIN:EDGE:SLOPE, 2–146
  - TRIGGER:MAIN:EDGE:SOURCE, 2–146
  - TRIGGER:MAIN:HOLDOFF?, 2–147
  - TRIGGER:MAIN:HOLD-OFF:VALUE, 2–148
  - TRIGGER:MAIN:LEVEL, 2–148
  - TRIGGER:MAIN:MODE, 2–149
  - TRIGGER:MAIN:TYPE, 2–150
  - TRIGGER:MAIN:VIDEO:FIELD, 2–150
  - TRIGGER:MAIN:VIDEO:POLARITY, 2–151
  - TRIGGER:MAIN:VIDEO:SOURCE, 2–152
  - TRIGGER:STATE?, 2–153
- TRIGGER:MAIN, 2–143
- TRIGGER:MAIN:EDGE?, 2–144
- TRIGGER:MAIN:EDGE:COUPLING, 2–145
- TRIGGER:MAIN:EDGE:SLOPE, 2–146
- TRIGGER:MAIN:EDGE:SOURCE, 2–146
- TRIGGER:MAIN:HOLDOFF?, 2–147

TRIGGER:MAIN:HOLD-  
     OFF:VALUE, 2-148  
 TRIGGER:MAIN:LEVEL, 2-148  
 TRIGGER:MAIN:MODE, 2-149  
 TRIGGER:MAIN:TYPE, 2-150  
 TRIGGER:MAIN:VIDEO:FIELD,  
     2-150  
 TRIGGER:MAIN:VIDEO:PO-  
     LARITY, 2-151  
 TRIGGER:MAIN:VID-  
     EO:SOURCE, 2-152  
 TRIGGER:STATE?, 2-153  
 \*TST? query, 2-154

## U

UNLOCK, 2-155

## V

VERBOSE, 2-155  
 Vertical bar cursors, 2-60  
 Vertical command group, 2-25  
 Vertical commands  
     CH<x>?, 2-46  
     CH<x>:BANDWIDTH, 2-47  
     CH<x>:COUPLING, 2-48  
     CH<x>:INVERT, 2-49  
     CH<x>:POSITION, 2-49  
     CH<x>:PROBE, 2-51  
     CH<x>:SCALE, 2-52  
     CH<x>:VOLTS, 2-53  
     MATH?, 2-109  
     MATH:DEFINE, 2-109  
     MATH:FFT:VERTICAL:POSI-  
         TION, 2-113  
     MATH:FFT:VERTICAL:SCALE,  
         2-113  
     SELECT?, 2-138  
     SELECT:<wfm>, 2-138

## W

\*WAI, 2-156  
 Wait for operation complete, 2-156  
 Waveform, command mnemonic,  
     2-9  
 Waveform command group, 2-26  
 Waveform commands  
     CURVE, 2-64  
     DATA, 2-65  
     DATA:DESTINATION, 2-66  
     DATA:ENCDG, 2-67  
     DATA:SOURCE, 2-69  
     DATA:START, 2-70  
     DATA:STOP, 2-71  
     DATA:TARGET, 2-72  
     DATA:WIDTH, 2-73  
     WAVFRM?, 2-157  
     WFMPRE?, 2-157  
     WFMPRE:<wfm>?, 2-176  
     WFMPRE:<wfm>:PT\_FMT,  
         2-178  
     WFMPRE:<wfm>:PT\_OFF,  
         2-179  
     WFMPRE:<wfm>:WFID, 2-179  
     WFMPRE:<wfm>:XINCR,  
         2-180  
     WFMPRE:<wfm>:XUNIT,  
         2-181, 2-182  
     WFMPRE:<wfm>:XZERO,  
         2-183  
     WFMPRE:<wfm>:YMULT,  
         2-184  
     WFMPRE:<wfm>:YOFF, 2-185  
     WFMPRE:<wfm>:YUNIT,  
         2-186, 2-187  
     WFMPRE:<wfm>:YZERO,  
         2-188  
     WFMPRE:BIT\_NR, 2-158  
     WFMPRE:BN\_FMT, 2-159  
     WFMPRE:BYT\_NR, 2-160

- WFMPRE:BYT\_OR, 2–160
- WFMPRE:ENCDG, 2–161
- WFMPRE:NR\_PT?, 2–162, 2–177
- WFMPRE:PT\_FMT, 2–163
- WFMPRE:PT\_OFF, 2–164
- WFMPRE:WFID, 2–165
- WFMPRE:XINCR, 2–166, 2–168
- WFMPRE:XMULT, 2–176
- WFMPRE:XOFF, 2–176
- WFMPRE:XUNIT, 2–167
- WFMPRE:XZERO, 2–169
- WFMPRE:YMULT, 2–170
- WFMPRE:YOFF, 2–171
- WFMPRE:YUNIT, 2–172, 2–173
- WFMPRE:YZERO, 2–174, 2–175, 2–189
- WFMPRE:ZMULT, 2–176
- WFMPRE:ZOFF, 2–176
- WFMPRE:ZUNIT, 2–176
- Waveform preamble, 2–31
- WAVFRM?, 2–157
- <wfm>, command mnemonic, 2–9
- WFMPRE?, 2–157
- WFMPRE:<wfm>?, 2–176
- WFMPRE:<wfm>:PT\_FMT, 2–178
- WFMPRE:<wfm>:PT\_OFF, 2–179
- WFMPRE:<wfm>:WFID, 2–179
- WFMPRE:<wfm>:XINCR, 2–180
- WFMPRE:<wfm>:XUNIT, 2–181, 2–182
- WFMPRE:<wfm>:XZERO, 2–183
- WFMPRE:<wfm>:YMULT, 2–184
- WFMPRE:<wfm>:YOFF, 2–185
- WFMPRE:<wfm>:YUNIT, 2–186, 2–187
- WFMPRE:<wfm>:YZERO, 2–188
- WFMPRE:BIT\_NR, 2–158
- WFMPRE:BN\_FMT, 2–159
- WFMPRE:BYT\_NR, 2–160
- WFMPRE:BYT\_OR, 2–160
- WFMPRE:ENCDG, 2–161
- WFMPRE:NR\_PT?, 2–162, 2–177
- WFMPRE:PT\_FMT, 2–163
- WFMPRE:PT\_OFF, 2–164
- WFMPRE:WFID, 2–165
- WFMPRE:XINCR, 2–166, 2–168
- WFMPRE:XMULT, 2–176
- WFMPRE:XOFF, 2–176
- WFMPRE:XUNIT, 2–167
- WFMPRE:XZERO, 2–169
- WFMPRE:YMULT, 2–170
- WFMPRE:YOFF, 2–171
- WFMPRE:YUNIT, 2–172, 2–173
- WFMPRE:YZERO, 2–174, 2–175, 2–189
- WFMPRE:ZMULT, 2–176
- WFMPRE:ZOFF, 2–176
- WFMPRE:ZUNIT, 2–176