Programmer Manual

Tektronix

TDS Family Digitizing Oscilloscopes

070-9702-00
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

## Getting Started

Overview of the Manual ................................................................. 1–1  
Setting Up Remote Communications ............................................ 1–3

## Syntax and Commands

Command Syntax ................................................................. 2–1  
Command and Query Structure ............................................... 2–1  
Clearing the TDS Family Oscilloscope ...................................... 2–3  
Command Entry ................................................................. 2–3  
Constructed Mnemonics ...................................................... 2–6  
Argument Types ............................................................... 2–7  
Syntax Diagrams ............................................................... 2–10  
**Command Groups** ............................................................ 2–11  
Acquisition Commands ....................................................... 2–11  
Alias Commands .............................................................. 2–12  
Application Menu Commands ............................................... 2–12  
Calibration and Diagnostic Commands ...................................... 2–13  
Cursor Commands .............................................................. 2–13  
Display Commands ............................................................. 2–14  
File System Commands ....................................................... 2–16  
Hardcopy Commands .......................................................... 2–17  
Horizontal Commands ......................................................... 2–17  
Limit Test Commands .......................................................... 2–19  
Measurement Commands ...................................................... 2–19  
Miscellaneous Commands .................................................... 2–21  
RS-232 Commands .............................................................. 2–22  
Save and Recall Commands .................................................. 2–22  
Status and Error Commands .................................................. 2–23  
Trigger Commands ............................................................. 2–24  
Vertical Commands ............................................................ 2–31  
Waveform Commands .......................................................... 2–32  
Zoom Commands ............................................................... 2–37  
**Command Descriptions** ...................................................... 2–39

## Status and Events

Registers ............................................................... 3–1  
Queues ................................................................. 3–5  
Event Handling Sequence ................................................... 3–6  
Synchronization Methods .................................................... 3–7  
Messages ................................................................. 3–12

## Programming Examples

Compiling the Example Programs ............................................. 4–2
Appendices

Appendix A: Character Charts ........................................ A–1
Appendix B: Reserved Words ............................................ B–1
Appendix C: Interface Specifications ................................. C–1
  GPIB Function Subsets ................................................ C–1
  Interface Messages ...................................................... C–2
Appendix D: Factory Initialization Settings ......................... D–1

Glossary and Index
List of Figures

Figure 1–1: Common Message Elements ........................................ 1–1
Figure 1–2: Functional Groupings and an Alphabetical List of
Commands ................................................................. 1–2
Figure 1–3: Service Requests (SRQs) Provide for Event (Interrupt)
Driven Programs ........................................................... 1–2
Figure 1–4: The Disks That Accompany This Manual .............. 1–3
Figure 1–5: GPIB Connector Location ............................. 1–3
Figure 1–6: How to Stack GPIB Connectors ...................... 1–4
Figure 1–7: Typical GPIB Network Configurations ............ 1–5
Figure 1–8: Selecting the I/O System in the Main Menu .......... 1–5
Figure 1–9: Selecting the GPIB Address in the GPIB Configuration
Side Menu ................................................................. 1–6

Figure 2–1: Command Message Elements ............................. 2–2
Figure 2–2: Block Argument Example ............................. 2–9
Figure 2–3: Typical Syntax Diagrams .............................. 2–10
Figure 2–4: Message Window Coordinates ...................... 2–198
Figure 2–5: LESSThan and MOREThan Arguments ........... 2–240

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

Figure 4–1: Equipment Needed to Run the Example Programs .... 4–1
List of Tables

Table 2–1: BNF Symbols and Meanings ........................................ 2–1
Table 2–2: Command Message Elements ................................. 2–2
Table 2–3: Comparison of Header Off and On Responses .......... 2–3
Table 2–4: Acquisition Commands ................................. 2–11
Table 2–5: Alias Commands ............................................. 2–12
Table 2–6: Application Menu Commands ............................. 2–12
Table 2–7: Calibration and Diagnostic Commands ................. 2–13
Table 2–8: Cursor Commands ........................................... 2–13
Table 2–9: Display Commands ......................................... 2–14
Table 2–10: File System Commands .................................... 2–16
Table 2–11: Hardcopy Commands ...................................... 2–17
Table 2–12: Horizontal Commands .................................... 2–17
Table 2–13: Limit Test Commands ..................................... 2–19
Table 2–14: Measurement Commands .................................. 2–20
Table 2–15: Miscellaneous Commands ................................ 2–21
Table 2–16: RS-232 Commands ........................................ 2–22
Table 2–17: Save and Recall Commands ............................ 2–23
Table 2–18: Status and Error Commands ............................ 2–23
Table 2–19: Trigger Commands ....................................... 2–25
Table 2–20: Vertical Commands ....................................... 2–31
Table 2–21: Waveform Commands .................................... 2–35
Table 2–22: Zoom Commands .......................................... 2–37
Table 2–23: Waveform Data Points Supported for Reference Locations ........................................... 2–54
Table 2–24: Commands that Affect BUSY? Response .............. 2–60
Table 2–25: Offset Ranges for the TDS 400A, 510A, 540B, 600B, & 700A (All Channels) and the TDS 520B, 724A, & 782A (Channel 1 & Channel 2) using a 1x Probe ....................... 2–65
Table 2–26: Offset Ranges for the TDS 520B, 724A, & 782A (Aux 1 & Aux 2) using a 1x Probe ................................. 2–65
Table 2–27: DATa and WFMPre Parameter Settings ............... 2–91
Table 2–28: XY Format Pairs .......................................... 2–113
Table 2–29: Record Length Values (<NR1>) ......................... 2–155
Table 2–30: Commands that Generate an Operation Complete Message ................................................ 2–202
Table 2–31: Additional WFMPre Commands ........................ 2–298
Table 3–1: SESR Bit Functions ........................................ 3–2
Table 3–2: SBR Bit Functions ........................................ 3–3
Table 3–3: No Event Messages ........................................ 3–12
Table 3–4: Command Error Messages – CME Bit 5 ............ 3–12
Table 3–5: Execution Error Messages – EXE Bit 4 ............. 3–14
Table 3–6: Device Error Messages – DDE Bit 3 ................. 3–17
Table 3–7: System Event Messages ................................. 3–17
Table 3–8: Execution Warning Messages – EXE Bit 4 ......... 3–18
Table 3–9: Internal Warning Messages ............................ 3–19
Table A–1: The TDS Character Set ................................. A–1
Table A–2: ASCII & GPIB Code Chart ............................. A–2
Table C–1: TDS Family Oscilloscope Standard Interface Message ............................. C–2
Table D–1: Factory Initialization Settings .......................... D–1
Preface


Related Manuals


Table i: Other Documentation

<table>
<thead>
<tr>
<th>Manual</th>
<th>Tek Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TDS User Manual</strong></td>
<td></td>
</tr>
<tr>
<td>TDS 420A, 430A, 460A, &amp; 510A</td>
<td>070–9701–XX</td>
</tr>
<tr>
<td>TDS 500B, 600B, &amp; 700A</td>
<td>070–9719–XX</td>
</tr>
<tr>
<td>TDS 500A/B &amp; 600A/B Option 05 Video Trigger</td>
<td>070–8748–XX</td>
</tr>
<tr>
<td><strong>TDS Reference</strong></td>
<td></td>
</tr>
<tr>
<td>TDS 420A, 430A, &amp; 460A</td>
<td>070–9700–XX</td>
</tr>
<tr>
<td>TDS 500B, 600B, &amp; 700A</td>
<td>070–9382–XX</td>
</tr>
<tr>
<td><strong>TDS Performance Verification</strong></td>
<td></td>
</tr>
<tr>
<td>TDS 420A, 430A, &amp; 460A</td>
<td>070–9705–XX</td>
</tr>
<tr>
<td>TDS 510A</td>
<td>070–9706–XX</td>
</tr>
<tr>
<td>TDS 500B, 600B, &amp; 700A</td>
<td>070–9720–XX</td>
</tr>
<tr>
<td><strong>TDS Service Manual</strong></td>
<td></td>
</tr>
<tr>
<td>TDS 420A, 430A, &amp; 460A</td>
<td>070–9703–XX</td>
</tr>
<tr>
<td>TDS 510A</td>
<td>070–9704–XX</td>
</tr>
<tr>
<td>TDS 500B, 600B, &amp; 700A</td>
<td>070–9721–XX</td>
</tr>
</tbody>
</table>
Getting Started

You can write computer programs that remotely set the oscilloscope front panel controls or that take measurements and read those measurements for further analysis or storage.

To help you get started with programming the oscilloscope, this section includes the following sections:

- **Overview of the Manual** – summarizes the type of programming information contained in each major section of this manual.
- **Setting Up Remote Communications** – describes how to physically connect the oscilloscope to a controller and set the appropriate front panel controls.

Overview of the Manual

The information contained in each major section of this manual is described below.

**Syntax and Commands**

The *Syntax and Commands* chapter describes the structure and content of the messages your program sends to the digitizing oscilloscope. Figure 1–1 shows a syntax diagram and command parts as described in the *Command Syntax* subsection.

![Syntax Diagram](image)

**Figure 1–1: Common Message Elements**

Chapter 2 also describes the effect of each command and provides examples of how you might use it. The *Command Groups* section provides a list by functional area. The *Command Descriptions* section arranges commands alphabetically (Figure 1–2).
The program may request information from the oscilloscope. The oscilloscope provides information in the form of status and error messages. Figure 1–3 illustrates the basic operation of this system.

The Status and Events chapter starting on page 3–1 describes how to use service requests (SRQs) and various event messages in your programs.

Your program requests status and event reports.

TDS sends status and event reports.

The Programming Examples chapter starting on page 4–1 describes some example digitizing oscilloscope programs and how to compile them. The disks that come with this manual (Figure 1–4) have an executable and a Microsoft QuickBASIC 4.5 and a Microsoft QuickC 2.5 source-code version of each program.
Setting Up Remote Communications

Even the best instrument control program will not do much if the instrument is not connected to the controller.

The digitizing oscilloscope has a 24-pin GPIB connector on its rear panel, as shown in Figure 1–5. This connector has a D-type shell and conforms to IEEE Std 488.1–1987.

Attach an IEEE Std 488.1–1987 GPIB cable (available from Tektronix as part number 012–0991–00) to this connector.
If needed, you can stack GPIB connectors as shown in Figure 1–6.

![Image of GPIB connectors stacked]

**Figure 1–6: How to Stack GPIB Connectors**

**GPIB Requirements**

Observe these rules when you use your digitizing oscilloscope with a GPIB network:

- Assign a unique device address to each device on the bus. No two devices can share the same device address.
- Do not connect more than 15 devices to any one bus.
- Connect one device for every 2 meters (6 feet) of cable used.
- Do not use more than 20 meters (65 feet) of cable to connect devices to a bus.
- Turn on at least two-thirds of the devices on the network while using the network.
- Connect the devices on the network in a star or linear configuration as shown in Figure 1–7. Do not use loop or parallel configurations.
You need to set the GPIB parameters of the digitizing oscilloscope to match the configuration of the bus. Once you have set these parameters, you can control the digitizing oscilloscope through the GPIB interface.

1. Press the **UTILITY (SHIFT DISPLAY)** button to display the Utility menu.

2. Press the **System** button in the main menu until it highlights the **I/O** selection in the pop-up menu. See Figure 1–8.
3. Press the **Port** button in the main menu until it highlights the **GPIB** selection in the pop-up menu. See Figure 1–9.

4. Press the **Configure** button in the main menu to display the **GPIB Configuration** side menu. See Figure 1–9.

5. Press the **Talk/Listen Address** side menu button, and set the **GPIB address** using either the general purpose knob or, if available, the keypad.

![Figure 1–9: Selecting the GPIB Address in the GPIB Configuration Side Menu](image)

The digitizing oscilloscope is set up for bidirectional communication with your controller. If you wish to isolate the digitizing oscilloscope from the bus:

- Press the **Off Bus** side menu button. This disables all communication with the controller.

If you wish to enter a special mode of operation to communicate directly with non-488.2 hard copy devices:

- Press the **Hardcopy** side menu button to have the digitizing oscilloscope send hard copy information only when you press the **HARDCOPY** button (and accept a **HARDCOPY ABORT** command).
Command Syntax

You can control the digitizing oscilloscope through the GPIB interface using commands and queries. This section describes the syntax these commands and queries use. It also describes the conventions the digitizing oscilloscope uses to process them. The next section, entitled Command Groups, lists the commands and queries themselves.

You transmit commands to the digitizing oscilloscope using the enhanced American Standard Code for Information Interchange (ASCII) character encoding. Appendix A: Character Charts on page A–2 contains a chart of the ASCII character set.

This manual describes commands and queries using Backus-Naur Form (BNF) notation and syntax diagrams.

This manual uses the BNF symbols shown in Table 2–1 below:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;  &gt;</td>
<td>Defined element</td>
</tr>
<tr>
<td>::=</td>
<td>Is defined as</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>{  }</td>
<td>Group; one element is required</td>
</tr>
<tr>
<td>[  ]</td>
<td>Optional; can be omitted</td>
</tr>
<tr>
<td>. . .</td>
<td>Previous element(s) may be repeated</td>
</tr>
<tr>
<td>(  )</td>
<td>Comment</td>
</tr>
</tbody>
</table>

Command and Query Structure

Commands consist of set commands and query commands (usually simply called commands and queries). Commands modify instrument settings or tell the digitizing oscilloscope to perform a specific action. Queries cause the digitizing 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 differs from the set form by its question mark on the end. For example, the set command ACQuire:M0DE has a query form ACQuire:M0DE?.

Not all commands have both a set and a query form. Some commands have set only and some have query only.
A command message is a command or query name followed by any information the digitizing oscilloscope needs to execute the command or query. Command messages may contain five element types, defined in Table 2–2 and shown in the example in Figure 2–1.

Table 2–2: Command Message Elements

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

Figure 2–1: Command Message Elements

**Commands**  
Commands have the structure:

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

**Queries**

Queries 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, `MEASURE:MEAS<x>:DELAY:DIRection?` returns the starting point and direction of the edge of a delayed measurement, while `MEASURE:MEAS<x>:DELAY?` returns the current settings of all delayed measurement parameters, and `MEASURE:MEAS<x>?` returns all the measurement parameters for the specified measurement.

**Headers in Query Responses**

You can control whether the digitizing oscilloscope returns headers as part of the query response. Use the `HEADER` command to control this feature. If header is on, the query response returns command headers and formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. Table 2–3 shows the difference in responses.

<table>
<thead>
<tr>
<th>Query</th>
<th>Header Off Response</th>
<th>Header On Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPMenu:TITLE?</td>
<td>&quot;Test Setup&quot;</td>
<td>:APPMenu:TITLE &quot;Test Setup&quot;</td>
</tr>
<tr>
<td>ACQuire:NUMAVg?</td>
<td>100</td>
<td>:ACQUIRE:NUMAVG 100</td>
</tr>
</tbody>
</table>

**Clearing the TDS Family Oscilloscope**

You can clear the Output Queue and reset the digitizing oscilloscope to accept a new command or query by using the Device Clear (DCL) GPIB command.

**Command Entry**

The following rules apply:
You can enter commands in upper or lower case.

You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).

The digitizing oscilloscope ignores commands consisting of any combination of white space characters and line feeds.

**Abbreviating Commands**

You can abbreviate many digitizing oscilloscope commands. Each command listing in the *Commands* section shows the minimum acceptable abbreviations in capitals. For example, you can enter the command **ACQuire:NUMAvg** simply as **ACQ:NUMA** or **acq:numa**.

**NOTE.** Keep in mind that abbreviation rules change over time as new TDS models get introduced. Thus, for the most robust code, use the full spelling. Avoid using the command abbreviations.

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

**Concatenating Commands**

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

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

1. Separate completely different headers by a semicolon and by the beginning colon on all commands but the first. For example, the commands **TRIGger:MODE NORMAL** and **ACQuire:NUMAvg 10** would be concatenated into a single command:

```
TRIGger:MODE NORMAL;ACQuire:NUMAvg 10
```

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands **ACQuire:MODE ENvelope** and **ACQuire:NUMAvg 10** into a single command:

```
ACQuire:MODE ENvelope; NUMAvg 10
```

The longer version works equally well:

```
ACQuire:MODE ENvelope;ACQuire:NUMAvg 10
```

3. Never precede a star (*) command with a colon:
ACQuire:MODE ENvelope;*TRG

Any commands that follow will be processed as if the star command was not there so

ACQuire:MODE ENvelope;*TRG;NUMAVg 10

will set the acquisition mode to envelope and set the number of acquisitions for averaging to 10.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display intensity for text is 80% and for the waveform it is 90%, the concatenated query

DISplay:INTEnsity:TEXT?;WAVEform?

will return either :DISPlay:INTENSITY:TEXT 80;:DISPlay:INTENSITY:WAVEFORM 90 if header is on or 80;90 if header is off.

5. Set commands and queries may be concatenated in the same message. For example,

ACQuire:MODE NORMAL;NUMAVg?;STATE?

is a valid message that sets the acquisition mode to normal. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

Here are some invalid concatenations:

- DISPlay:INTEnsity:TEXT 80;ACQuire:NUMAVg 10 (no colon before ACQuire)
- DISPlay:INTEnsity:TEXT 80;:WAVEform 90 (extra colon before WAVEform — could use DISPlay:INTEnsity:WAVEform instead)
- DISPlay:INTEnsity:TEXT 80;:*TRG (colon before a star (*) command)
- APPMenu:LABel:BOTTOM1 "foo";LABel:BOTTOM2 "fee" (levels of the mnemonics are different — either remove the second use of LABel: or place :APPMenu: in front of LABel:BOTTOM2)

**Message Terminators**

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;EOM&gt;</td>
<td>Message terminator</td>
</tr>
</tbody>
</table>
The end-of-message terminator may be the END message (EOI asserted concurrently with the last data byte), the ASCII code for line feed (LF) sent as the last data byte, or both. The digitizing oscilloscope always terminates messages with LF and EOI. It allows white space before the terminator. For example, it allows CR LF.

### Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be either CH1, CH2, CH3, or CH4. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:VOLT command, and there is also a CH2:VOLT command. In the command descriptions, this list of choices is abbreviated as CH<x>.

#### Application Menu Mnemonics

When the application menu is displayed, commands may specify which menu button to use.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOTTOM&lt;x&gt;</td>
<td>A main menu button selector; &lt;x&gt; is 1, 2, 3, 4, 5, 6, or 7. Main menu buttons are located along the bottom of the display and are numbered left to right, starting with 1.</td>
</tr>
<tr>
<td>RIGHT&lt;x&gt;</td>
<td>A side menu button selector; &lt;x&gt; is 1, 2, 3, 4, or 5. Side menu buttons are located along the right side of the display and are numbered top to bottom, starting with 1.</td>
</tr>
</tbody>
</table>

#### Cursor Position Mnemonics

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSITION&lt;x&gt;</td>
<td>A cursor selector; &lt;x&gt; is either 1 or 2.</td>
</tr>
</tbody>
</table>

#### Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. Up to four automated measurements may be displayed with each displayed waveform. The displayed measurements are specified in this way:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAS&lt;x&gt;</td>
<td>A measurement specifier; &lt;x&gt; is either 1 (top), 2, 3, or 4 (bottom).</td>
</tr>
</tbody>
</table>
Commands specify the channel to use as a mnemonic in the header.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;</td>
<td>A channel specifier; &lt;x&gt; is either 1, 2, 3, or 4.</td>
</tr>
<tr>
<td></td>
<td>For the TDS 520B, 620B, 680B, 724A, and 782A, CH3 and CH4 represent the front-panel inputs labeled AUX 1 and AUX 2 respectively.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH&lt;x&gt;</td>
<td>A math waveform specifier; &lt;x&gt; is 1, 2, or 3.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF&lt;x&gt;</td>
<td>A reference waveform specifier; &lt;x&gt; is either 1, 2, 3, or 4.</td>
</tr>
</tbody>
</table>

In some commands, you can specify a waveform regardless of whether it is a channel waveform, a math waveform, or a reference waveform. Specify such a waveform as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;wfm&gt;</td>
<td>Can be CH&lt;x&gt;, MATH&lt;x&gt; or REF&lt;x&gt;</td>
</tr>
</tbody>
</table>

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

Many digitizing oscilloscope commands require numeric arguments. The syntax shows the format that the digitizing oscilloscope returns in response to a query.
This is also the preferred format when sending the command to the digitizing oscilloscope though any of the formats will be accepted. This manual represents these arguments as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NR1&gt;</td>
<td>Signed integer value</td>
</tr>
<tr>
<td>&lt;NR2&gt;</td>
<td>Floating point value without an exponent</td>
</tr>
<tr>
<td>&lt;NR3&gt;</td>
<td>Floating point value with an exponent</td>
</tr>
</tbody>
</table>

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

**Quoted String Arguments**

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

"this is a quoted string"

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;QString&gt;</td>
<td>Quoted string of ASCII text</td>
</tr>
</tbody>
</table>

Follow these rules when you use quoted strings:

1. A quoted string can include any character defined in the 7-bit ASCII character set. (See Appendix A: Character Charts on page A–2).
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 imbedded 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 invalid strings:

- "Invalid string argument'
  (quotes are not of the same type)
- "test<EOI>"
  (termination character is embedded in the string)

Block Arguments

Several digitizing oscilloscope commands use a block argument form:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NZDig&gt;</td>
<td>A nonzero digit character, in the range 1–9</td>
</tr>
<tr>
<td>&lt;Dig&gt;</td>
<td>A digit character, in the range 0–9</td>
</tr>
<tr>
<td>&lt;DChar&gt;</td>
<td>A character with the hex equivalent of 00 through FF hexadecimal (0 through 255 decimal)</td>
</tr>
<tr>
<td>&lt;Block&gt;</td>
<td>A block of data bytes, defined as:</td>
</tr>
<tr>
<td></td>
<td>&lt;Block&gt; ::= #&lt;NZDig&gt;&lt;Dig&gt;[=&lt;Dig&gt;...] [&lt;DChar&gt;...]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<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.

Figure 2–2 provides a diagram of block argument use.
Syntax Diagrams

The syntax diagrams in this manual use the following symbols and notation:

- Circles and ovals contain literal elements. You must send most elements exactly as shown. The command mnemonics are shown in both upper and lower case to distinguish between complete and abbreviated spellings. These elements are not case sensitive. You can omit the lower case portion of the mnemonic.

- Boxes contain the defined elements described earlier in this section, such as `<NR3>` or `<String>`.

- Elements are connected by arrows that show the allowed paths through the diagram and, thus, the orders in which you can send the elements. Parallel paths show that you must take one and only one of the paths. A path around a group of elements shows that those elements are optional. Loops show elements that you can repeat.

Figure 2–3 shows the structure of a few typical syntax diagrams.

![Figure 2–3: Typical Syntax Diagrams](image-url)
Command Groups

This section lists TDS Family Oscilloscope commands in two ways. It first presents them by functional groups. It then lists them alphabetically. The functional group list starts below. The alphabetical list provides more detail on each command and starts on page 2–39.

The TDS Family Oscilloscope GPIB interface conforms to Tektronix standard codes and formats and IEEE Std 488.2–1987 except where noted.

Acquisition Commands

Acquisition commands affect waveform acquisition. These commands control mode, averaging, enveloping, and single-waveform acquisition. (Persistence controls are in the Display Commands section on page 2–14.) Table 2–4 lists these commands.

Table 2–4: Acquisition Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE?</td>
<td>Return acquisition parameters</td>
</tr>
<tr>
<td>ACQUIRE:AUTOAVe</td>
<td>Save waveforms to reference memory</td>
</tr>
<tr>
<td>(TDS 500B, 600B &amp; 700A)</td>
<td></td>
</tr>
<tr>
<td>ACQUIRE:MODE</td>
<td>Acquisition mode</td>
</tr>
<tr>
<td>ACQUIRE:NMACq?</td>
<td>Return # of acquisitions obtained</td>
</tr>
<tr>
<td>ACQUIRE:NUMAVg</td>
<td>Number of acquisitions for average</td>
</tr>
<tr>
<td>ACQUIRE:NUMEnv</td>
<td>Number of acquisitions for envelope</td>
</tr>
<tr>
<td>ACQUIRE:REPET (TDS 400A, 510A, 500B, &amp; 700A)</td>
<td>Repetitive acquisition mode</td>
</tr>
<tr>
<td>ACQUIRE:STATE</td>
<td>Start or stop acquisition system</td>
</tr>
<tr>
<td>ACQUIRE:STOPAfter</td>
<td>Acquisition control</td>
</tr>
</tbody>
</table>
Alias Commands

Alias commands let you define your own commands as a sequence of standard commands. This is useful when you use the same commands each time you perform a certain task, such as setting up measurements. Table 2–5 lists these commands.

Table 2–5: Alias Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIas</td>
<td>Turn the alias state on and off</td>
</tr>
<tr>
<td>ALIas:CATalog?</td>
<td>Return a list of aliases</td>
</tr>
<tr>
<td>ALIas:DEFINE</td>
<td>Create a new alias</td>
</tr>
<tr>
<td>ALIas:DELETE</td>
<td>Remove an alias</td>
</tr>
<tr>
<td>ALIas:DELETE:ALL</td>
<td>Remove all aliases</td>
</tr>
<tr>
<td>ALIas:DELETE:NAME</td>
<td>Remove a named alias</td>
</tr>
<tr>
<td>ALIas:STATE</td>
<td>Turn the alias state on and off</td>
</tr>
</tbody>
</table>

Application Menu Commands

Application menu commands let you define special-purpose menus. You can define labels for the main and side menus as well as a side menu title. You can display an Application menu by either pressing the front-panel APPLICATION button or sending the APPMenu ACTivate command. Table 2–6 lists these commands.

When the digitizing oscilloscope displays an Application menu and a user presses a front-panel button, the oscilloscope generates an event that tells the controller which button the user pressed. You can also set up the event reporting system so that it generates a Service Request when a user presses a menu button.

Table 2–6: Application Menu Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPMenu</td>
<td>Display the application menu</td>
</tr>
<tr>
<td>APPMenu:LABel</td>
<td>Return or remove all application menu button labels</td>
</tr>
<tr>
<td>APPMenu:LABel:BOTTOM&lt;x&gt;</td>
<td>Label for a bottom menu button</td>
</tr>
<tr>
<td>APPMenu:LABel:RIGHT&lt;x&gt;</td>
<td>Label for a side menu button</td>
</tr>
<tr>
<td>APPMenu:TITLe</td>
<td>Create a title for the application menu</td>
</tr>
</tbody>
</table>
Calibration and Diagnostic Commands

Calibration and Diagnostic commands let you start the self-calibration and diagnostic routines that are built into the oscilloscope. The diagnostic test operation includes selecting the test sequence, executing the sequence, and viewing the results. Table 2–7 lists these commands.

Table 2–7: Calibration and Diagnostic Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL?</td>
<td>Perform an internal self-calibration</td>
</tr>
<tr>
<td>DIAg:RESULT:FLAG?</td>
<td>Return diagnostic tests status</td>
</tr>
<tr>
<td>DIAg:RESULT:LOG?</td>
<td>Return diagnostic test sequence results</td>
</tr>
<tr>
<td>DIAg:SELEc:ACQUSition</td>
<td>Acquisition system diagnostic test sequence</td>
</tr>
<tr>
<td>DIAg:SELEc:ALl</td>
<td>Diagnostic test sequence for Acquisition, Processor, Display, and Front panel</td>
</tr>
<tr>
<td>DIAg:SELEc:CPU</td>
<td>Processor diagnostic test sequence</td>
</tr>
<tr>
<td>DIAg:SELEc:DISplay</td>
<td>Display system diagnostic test sequence</td>
</tr>
<tr>
<td>DIAg:SELEc:FPAnel</td>
<td>Front panel diagnostic test sequence</td>
</tr>
<tr>
<td>DIAg:STATE</td>
<td>Control of diagnostic tests</td>
</tr>
</tbody>
</table>

Cursor Commands

Cursor commands provide control over cursor (caliper) display and readout. Table 2–8 lists these commands.

Table 2–8: Cursor Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSor?</td>
<td>Return cursor settings</td>
</tr>
<tr>
<td>CURSor:FUNction</td>
<td>Cursors on or off; select cursor type</td>
</tr>
<tr>
<td>CURSor:HBArs?</td>
<td>Return H bar settings</td>
</tr>
<tr>
<td>CURSor:HBArs:DELta?</td>
<td>Return distance between H bars</td>
</tr>
<tr>
<td>CURSor:HBArs:POSITION&lt;x&gt;</td>
<td>Position a horizontal cursor</td>
</tr>
<tr>
<td>CURSor:HBArs:POSITION&lt;x&gt;Pcnt (TDS 400A)</td>
<td>Position a horizontal cursor in units of % of vertical range</td>
</tr>
<tr>
<td>CURSor:HBArs:SELEct</td>
<td>Set which cursor the knob controls</td>
</tr>
<tr>
<td>CURSor:HBArs:UNIts</td>
<td>Set H bar units</td>
</tr>
<tr>
<td>CURSor:MODe</td>
<td>Set cursor tracking mode</td>
</tr>
</tbody>
</table>
## Cursor Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSor:PAIred</td>
<td>Positions paired cursors. Also, returns settings</td>
</tr>
<tr>
<td>CURSor:PAIred:HDELTA?</td>
<td>Return horizontal distance between 1st and 2nd paired cursors</td>
</tr>
<tr>
<td>CURSor:PAIred:HPOS?</td>
<td>Return horizontal position of 1st paired cursor</td>
</tr>
<tr>
<td>CURSor:PAIred:HPOS2?</td>
<td>Return horizontal position of 2nd paired cursor</td>
</tr>
<tr>
<td>CURSor:PAIred:POSITION&lt;x&gt;</td>
<td>Set or return vbar position of the 1st or 2nd paired cursor</td>
</tr>
<tr>
<td>CURSor:PAIred:POSITION&lt;x&gt;Pcnt (TDS 400A)</td>
<td>Position the horizontal paired cursor in units of % of record length</td>
</tr>
<tr>
<td>CURSor:PAIred:SELECT</td>
<td>Select active paired cursor</td>
</tr>
<tr>
<td>CURSor:PAIred:UNITS</td>
<td>Set paired cursor units</td>
</tr>
<tr>
<td>CURSor:PAIred:VDELTA?</td>
<td>Return vertical distance between 1st and 2nd paired cursors</td>
</tr>
<tr>
<td>CURSor:VBArs</td>
<td>Position vertical bar cursors</td>
</tr>
<tr>
<td>CURSor:VBArs:DELTa?</td>
<td>Return horizontal distance between cursors</td>
</tr>
<tr>
<td>CURSor:VBArs:POSITION&lt;x&gt;</td>
<td>Position a vertical cursor</td>
</tr>
<tr>
<td>CURSor:VBArs:POSITION&lt;x&gt;Pcnt (TDS 400A)</td>
<td>Position a vertical cursor in units of % of record length</td>
</tr>
<tr>
<td>CURSor:VBArs:SELECT</td>
<td>Set which cursor the knob controls</td>
</tr>
<tr>
<td>CURSor:VBArs:UNITS</td>
<td>Set vertical cursors to seconds, frequency, or to lines (with option 05 video on the TDS 510A, 500B, 600B, &amp; 700A)</td>
</tr>
<tr>
<td>CURSor:VBArs:UNITSTRING(TDS 400A)</td>
<td>Return unit string for the vertical bar cursor</td>
</tr>
</tbody>
</table>

## Display Commands

Display commands let you change the graticule style, change the displayed intensities, display messages, and clear the menu. Table 2–9 lists these commands.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEARMenu</td>
<td>Clear menus from display</td>
</tr>
<tr>
<td>DISplay?</td>
<td>Return display settings</td>
</tr>
<tr>
<td>DISplay:CLOCK</td>
<td>Control the display of the date/time stamp</td>
</tr>
</tbody>
</table>
### Table 2–9: Display Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY:COLOR:CONTRast (TDS 6X4B &amp; 700A)</td>
<td>Control collision contrast</td>
</tr>
<tr>
<td>DISPLAY:COLOR:MAP:&lt;item-name&gt;:=BYCONTents (TDS 6X4B &amp; 700A)</td>
<td>Set color for math or ref waveform to the color of the waveform content</td>
</tr>
<tr>
<td>DISPLAY:COLOR:MAP:&lt;item-name&gt;:TO (TDS 6X4B &amp; 700A)</td>
<td>Set color for math or ref waveform to specific color index</td>
</tr>
<tr>
<td>DISPLAY:COLOR:PALETTE:PERSISTence (TDS 6X4B &amp; 700A)</td>
<td>Set current persistence palette to a preset persistence palette</td>
</tr>
<tr>
<td>DISPLAY:COLOR:PALETTE:REGular (TDS 6X4B &amp; 700A)</td>
<td>Set current palette to a preset palette</td>
</tr>
<tr>
<td>DISPLAY:COLOR:PALETTE:RESETALL (TDS 6X4A &amp; 700A)</td>
<td>Reset all palettes to their factory default settings</td>
</tr>
<tr>
<td>DISPLAY:COLOR:PALETTE:&lt;palette name&gt;:RESET (TDS 6X4B &amp; 700A)</td>
<td>Reset a selected palette to its factory default settings</td>
</tr>
<tr>
<td>DISPLAY:COLOR:PALETTE:&lt;palette name&gt;:&lt;item name&gt; (TDS 6X4B &amp; 700A)</td>
<td>Set the color of a selected item on a selected palette</td>
</tr>
<tr>
<td>DISPLAY:FILTER</td>
<td>Displayed data interpolation</td>
</tr>
<tr>
<td>DISPLAY:FORMAT</td>
<td>YT or XY display</td>
</tr>
<tr>
<td>DISPLAY:GRATICule</td>
<td>Graticule style</td>
</tr>
<tr>
<td>DISPLAY:INSTavu:PERSISTence (TDS 500B &amp; 700A)</td>
<td>Set InstaVu persistence type – variable or infinite</td>
</tr>
<tr>
<td>DISPLAY:INSTavu:STYLE (TDS 500B &amp; 700A)</td>
<td>Set InstaVu waveform dots or vector style</td>
</tr>
<tr>
<td>DISPLAY:INSTavu:VARpersist (TDS 500B &amp; 700A)</td>
<td>Set InstaVu variable persistence decay time</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY?</td>
<td>Return intensity settings</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:CONTRast (TDS 4X0A, 510A, 5X0B, &amp; 6X0B)</td>
<td>Waveform intensified zone brightness</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:OVERALL (TDS 4X0A, 510A, 5X0B, &amp; 6X0B)</td>
<td>Set main brightness</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:TEXT</td>
<td>Set text brightness</td>
</tr>
<tr>
<td>DISPLAY:INTENSITY:WAVEform</td>
<td>Set waveform brightness</td>
</tr>
<tr>
<td>DISPLAY:MODE (TDS 500B &amp; 700A)</td>
<td>Set normal or InstaVu display mode</td>
</tr>
<tr>
<td>DISPLAY:PERSISTence</td>
<td>Set variable persistence decay time</td>
</tr>
</tbody>
</table>
Table 2–9: Display Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY:STYLE</td>
<td>Set waveform dots, vector, infinite persistence, or variable persistence</td>
</tr>
<tr>
<td>DISPLAY:TRIGBar</td>
<td>Control the display of the trigger bar/s on screen</td>
</tr>
<tr>
<td>DISPLAY:TRIGT</td>
<td>Control the display of the trigger indicator on screen</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>Remove text from the message window</td>
</tr>
<tr>
<td>MESSAGE:BOX</td>
<td>Set size and location of message window</td>
</tr>
<tr>
<td>MESSAGE:SHOW</td>
<td>Remove and display text in the message window</td>
</tr>
<tr>
<td>MESSAGE:STATE</td>
<td>Control display of message window</td>
</tr>
</tbody>
</table>

File System Commands

File system commands help you use the built-in 3.5 inch floppy disk drive (available with the File System). Table 2–10 lists these commands.

Table 2–10: File System Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESystem:COPY</td>
<td>Copy file to new file</td>
</tr>
<tr>
<td>FILESystem:CWD</td>
<td>Set directory path</td>
</tr>
<tr>
<td>FILESystem:DELETE</td>
<td>Delete named file</td>
</tr>
<tr>
<td>FILESystem:DELWarn</td>
<td>Set front-panel delete warning</td>
</tr>
<tr>
<td>FILESystem:DIR</td>
<td>Make directory</td>
</tr>
<tr>
<td>FILESystem:FORMAT</td>
<td>Format named drive</td>
</tr>
<tr>
<td>FILESystem:FREESpace</td>
<td>Return free space on current drive</td>
</tr>
<tr>
<td>FILESystem:MKDir</td>
<td>Make new directory</td>
</tr>
<tr>
<td>FILESystem:OVERWrite</td>
<td>Set file-overwrite protection</td>
</tr>
<tr>
<td>FILESystem:PRINT</td>
<td>Print file to port</td>
</tr>
<tr>
<td>FILESystem:REName</td>
<td>Assign new name to file</td>
</tr>
<tr>
<td>FILESystem:RMDir</td>
<td>Delete named directory</td>
</tr>
</tbody>
</table>
**Hardcopy Commands**

Hardcopy commands let you control the format of hardcopy output and the initiation and termination of hardcopies. Table 2–11 lists these commands.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARDCopy</td>
<td>Start or terminate hardcopy</td>
</tr>
<tr>
<td>HARDCopy:FILENAME (File System only)</td>
<td>Select file to send hardcopy data to</td>
</tr>
<tr>
<td>HARDCopy:FORMAT</td>
<td>Hardcopy output format</td>
</tr>
<tr>
<td>HARDCopy:LAYout</td>
<td>Hardcopy orientation</td>
</tr>
<tr>
<td>HARDCopy:PALETTE (TDS 6X4B &amp; 700A)</td>
<td>Select palette to use when making hardcopy</td>
</tr>
<tr>
<td>HARDCopy:PORT</td>
<td>Hardcopy port for output</td>
</tr>
</tbody>
</table>

**Horizontal Commands**

Horizontal commands control the time bases of the digitizing oscilloscope. You can set the time per division (or time per point) of both the main and delay time bases. You can also set the record lengths. Table 2–12 lists these commands.

You may substitute SECdiv for SCAle in the horizontal commands. This provides program compatibility with earlier models of Tektronix digitizing oscilloscopes.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HORIZontal?</td>
<td>Return horizontal settings</td>
</tr>
<tr>
<td>HORIZontal:CLOCKS (TDS 400A)</td>
<td>Enable internal or external clocks</td>
</tr>
<tr>
<td>HORIZontal:CLOCKS:MAXRate (TDS 400A)</td>
<td>Set maximum external clock rate</td>
</tr>
<tr>
<td>HORIZontal:DELAY?</td>
<td>Return delay time base settings</td>
</tr>
<tr>
<td>HORIZontal:DELAY:MODE</td>
<td>Delay time base mode</td>
</tr>
<tr>
<td>HORIZontal:DELAY:SCALE</td>
<td>Delay time base time per division</td>
</tr>
<tr>
<td>HORIZontal:DELAY:SECdiv</td>
<td>Same as HORIZontal:DELAY:SCALE</td>
</tr>
<tr>
<td>HORIZontal:DELAY:TIME</td>
<td>Delay time</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>HORizontal:DELay:TIME?</td>
<td>Return delay time parameters</td>
</tr>
<tr>
<td>HORizontal:DELay:TIME:RUNSAfter</td>
<td>Time to wait in delay-runs-after-main mode</td>
</tr>
<tr>
<td>HORizontal:DELay:TIME:TRIGAfter</td>
<td>Time to wait in delay-runs-after-trigger mode</td>
</tr>
<tr>
<td>HORizontal:FASTframe:COUNT (TDS 500B &amp; 700A)</td>
<td>Select FastFrame count</td>
</tr>
<tr>
<td>HORizontal:FASTframe:LENght (TDS 500B &amp; 700A)</td>
<td>Select length of each FastFrame frame</td>
</tr>
<tr>
<td>HORizontal:FASTframe:POSITION (TDS 500B &amp; 700A)</td>
<td>Select FastFrame frame to display</td>
</tr>
<tr>
<td>HORizontal:FASTframe:STATE (TDS 500B &amp; 700A)</td>
<td>Setup FastFrame acquisition</td>
</tr>
<tr>
<td>HORizontal:FITtoscreen</td>
<td>Setup waveform compress</td>
</tr>
<tr>
<td>HORizontal:MAIn?</td>
<td>Return main time per division</td>
</tr>
<tr>
<td>HORizontal:MAIn:SCAle</td>
<td>Main time base time per division</td>
</tr>
<tr>
<td>HORizontal:MAIn:SECdiv</td>
<td>Same as HORizontal:MAIn:SCAle</td>
</tr>
<tr>
<td>HORizontal:MODE</td>
<td>Turn delay time base on or off</td>
</tr>
<tr>
<td>HORizontal:POSITION</td>
<td>Portion of waveform to display</td>
</tr>
<tr>
<td>HORizontal:RECORDlength</td>
<td>Number of points in waveform record</td>
</tr>
<tr>
<td>HORizontal:ROLL (TDS 400A)</td>
<td>Set roll mode to auto or off</td>
</tr>
<tr>
<td>HORizontal:SCAle</td>
<td>Same as HORizontal:MAIn:SCAle</td>
</tr>
<tr>
<td>HORizontal:SECdiv</td>
<td>Same as HORizontal:MAIn:SCAle</td>
</tr>
<tr>
<td>HORizontal:TRIgger?</td>
<td>Return trigger position</td>
</tr>
<tr>
<td>HORizontal:TRIgger:POSITION</td>
<td>Main time base trigger position</td>
</tr>
</tbody>
</table>
Limit Test Commands

The Limit Test commands let you automatically compare each incoming waveform against a template waveform. You set an envelope of limits around a waveform and let the digitizing oscilloscope find the waveforms that fall outside those limits. When it finds such a waveform, it can generate a hardcopy, ring a bell, stop and wait for your input, or any combination of these actions. Table 2–13 lists these commands.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMIT:BEL1</td>
<td>Ring bell when limit exceeded</td>
</tr>
<tr>
<td>LIMIT:COMPARE:CH&lt;x&gt;</td>
<td>Template to compare waveform to</td>
</tr>
<tr>
<td>LIMIT:COMPARE:MATH&lt;x&gt; (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Template to compare math waveform to</td>
</tr>
<tr>
<td>LIMIT:HARDCopy</td>
<td>Make hardcopy when limit exceeded</td>
</tr>
<tr>
<td>LIMIT:STATE</td>
<td>Turn limit testing on or off</td>
</tr>
<tr>
<td>LIMIT:TEMPLATE</td>
<td>Template to compare waveform to</td>
</tr>
<tr>
<td>LIMIT:TEMPLATE:DESTination</td>
<td>Reference storage for template waveform</td>
</tr>
<tr>
<td>LIMIT:TEMPLATE:SOURCE</td>
<td>Template waveform source</td>
</tr>
<tr>
<td>LIMIT:TEMPLATE:TOLERance:HORIZONTAL</td>
<td>Tested waveform horizontal tolerance</td>
</tr>
<tr>
<td>LIMIT:TEMPLATE:TOLERance:VERTical</td>
<td>Tested waveform vertical tolerance</td>
</tr>
</tbody>
</table>

Measurement Commands

Measurement commands control the automated measurement system. Table 2–14 lists these commands.

Up to four automated measurements can be displayed on the 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 displayed measurements, the measurement commands let you specify a fifth measurement, IMMEd. The immediate measurement has no front-panel equivalent. Immediate measurements are never displayed. Because they are computed only when needed, immediate measurements slow the waveform update rate less than displayed measurements.

Whether you use displayed or immediate measurements, you use the \texttt{VALUE?} query to obtain measurement results.
Measurement commands can set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement readout. Other parameters, such as reference levels, have only one value, which applies to all measurements.

**Table 2–14: Measurement Commands**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASUREMENT?</td>
<td>Return all measurement parameters</td>
</tr>
<tr>
<td>MEASUREMENT:CLEARSNAPSHOT</td>
<td>Take down measurement snapshot</td>
</tr>
<tr>
<td>MEASUREMENT:GATING</td>
<td>Set or query measurement gating</td>
</tr>
<tr>
<td>MEASUREMENT:IMMEDE?</td>
<td>Return immediate measurement parameters</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY?</td>
<td>Return info on immediate delay measurement</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:DIRection</td>
<td>Search direction to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:EDGE1</td>
<td>Which waveform edge to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:DELAY:EDGE2</td>
<td>Which waveform edge to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:SOURCE1</td>
<td>Channel to take measurement from</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:SOURCE2</td>
<td>Second channel to take measurement from (delay or “to” channel)</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:TYPEe</td>
<td>The measurement to be taken</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:UNITS?</td>
<td>Return measurement units</td>
</tr>
<tr>
<td>MEASUREMENT:IMMED:VALUE?</td>
<td>Return measurement result</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;?</td>
<td>Return parameters on measurement</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:DELAY?</td>
<td>Return delay measurement parameters</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:DELAY:DIRection</td>
<td>Search direction to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE1</td>
<td>Which waveform edge to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE2</td>
<td>Which waveform edge to use for delay measurements</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SOURCE1</td>
<td>Channel to take measurement from</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:SOURCE2</td>
<td>Second channel to take measurement from (delay or “to” channel)</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:STATE</td>
<td>Turn measurement display on or off</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:TYPEe</td>
<td>The measurement to be taken</td>
</tr>
<tr>
<td>MEASUREMENT:MEAS&lt;x&gt;:UNITS?</td>
<td>Return units to use for measurement</td>
</tr>
</tbody>
</table>
Table 2–14: Measurement Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASurement:MEAS&lt;x&gt;:VALue?</td>
<td>Return measurement result</td>
</tr>
<tr>
<td>MEASurement:METHod</td>
<td>Method for calculating reference levels</td>
</tr>
<tr>
<td>MEASurement:REFLevel?</td>
<td>Return reference levels</td>
</tr>
<tr>
<td>MEASurement:REFLevel:ABSolute:HIGH</td>
<td>The top level for risetime (90% level)</td>
</tr>
<tr>
<td>MEASurement:REFLevel:ABSolute:LOW</td>
<td>The low level for risetime (10% level)</td>
</tr>
<tr>
<td>MEASurement:REFLevel:ABSolute:MID</td>
<td>Mid level for measurements</td>
</tr>
<tr>
<td>MEASurement:REFLevel:ABSolute:MID2</td>
<td>Mid level for delay measurements</td>
</tr>
<tr>
<td>MEASurement:REFLevel:METHod</td>
<td>Method to assign HIGH and LOW levels: either % or absolute volts</td>
</tr>
<tr>
<td>MEASurement:REFLevel:PERCent:HIGH</td>
<td>The top level for risetime (90% level)</td>
</tr>
<tr>
<td>MEASurement:REFLevel:PERCent:LOW</td>
<td>The low level for risetime (10% level)</td>
</tr>
<tr>
<td>MEASurement:REFLevel:PERCent:MID</td>
<td>Mid level for measurements</td>
</tr>
<tr>
<td>MEASurement:REFLevel:PERCent:MID2</td>
<td>Mid level for delay measurements</td>
</tr>
<tr>
<td>MEASurement:SNAPShot</td>
<td>Display measurement snapshot</td>
</tr>
</tbody>
</table>

Miscellaneous Commands

Miscellaneous commands do not fit into other categories. Table 2–15 lists these commands.

Several commands and queries are common to all 488.2–1987 devices on the GPIB bus. The 488.2–1987 standard defines them. They begin with a star (*) character.

Table 2–15: Miscellaneous Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSet</td>
<td>Automatic instrument setup</td>
</tr>
<tr>
<td>BEL1</td>
<td>Audio alert</td>
</tr>
<tr>
<td>DATE</td>
<td>Set date</td>
</tr>
<tr>
<td>*DDT</td>
<td>Define group execute trigger (GET)</td>
</tr>
<tr>
<td>FACTory</td>
<td>Reset to factory default</td>
</tr>
<tr>
<td>HDR</td>
<td>Same as HEADer</td>
</tr>
<tr>
<td>HEADer</td>
<td>Return command header with query</td>
</tr>
<tr>
<td>*IDN?</td>
<td>Identification</td>
</tr>
</tbody>
</table>
### Table 2–15: Miscellaneous Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*LRN?</td>
<td>Learn device setting</td>
</tr>
<tr>
<td>LOCK</td>
<td>Lock front panel (local lockout)</td>
</tr>
<tr>
<td>NEWpass</td>
<td>Change password for User Protected Data</td>
</tr>
<tr>
<td>PASSword</td>
<td>Access to change User Protected Data</td>
</tr>
<tr>
<td>REM</td>
<td>No action; remark only</td>
</tr>
<tr>
<td>SET?</td>
<td>Same as *LRN?</td>
</tr>
<tr>
<td>TEKSecure</td>
<td>Initialize waveforms and setups</td>
</tr>
<tr>
<td>TIME</td>
<td>Set time</td>
</tr>
<tr>
<td>*TRG</td>
<td>Perform Group Execute Trigger (GET)</td>
</tr>
<tr>
<td>*TST?</td>
<td>Self-test</td>
</tr>
<tr>
<td>UNLOCK</td>
<td>Unlock front panel (local lockout)</td>
</tr>
<tr>
<td>VERBose</td>
<td>Return full command name or minimum spellings with query</td>
</tr>
</tbody>
</table>

### RS-232 Commands

RS-232 commands allow you to utilize the serial communications port (available with the RS-232/Centronics Hardcopy Interface). Table 2–16 lists these commands.

### Table 2–16: RS-232 Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS232?</td>
<td>Return RS-232 parameters</td>
</tr>
<tr>
<td>RS232:BAUd</td>
<td>Set baud rate</td>
</tr>
<tr>
<td>RS232:HARDFlagging</td>
<td>Set hard flagging</td>
</tr>
<tr>
<td>RS232:PARity</td>
<td>Set parity</td>
</tr>
<tr>
<td>RS232:SOFTFlagging</td>
<td>Set soft flagging</td>
</tr>
<tr>
<td>RS232:STOPBits</td>
<td>Set # of stop bits</td>
</tr>
</tbody>
</table>

### Save and Recall Commands

Save and Recall commands allow you to store and retrieve internal waveforms and settings. When you “save a setup,” you save all the settings of the digitizing oscilloscope. When you then “recall a setup,” the digitizing oscilloscope restores
itself to the state it was in when you originally saved that setting. Table 2–17 lists these commands.

**Table 2–17: Save and Recall Commands**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL0cate?</td>
<td>Return number of allocated and unallocated data points</td>
</tr>
<tr>
<td>ALL0cate:WAVEform?</td>
<td>Return number of allocated data points</td>
</tr>
<tr>
<td>ALL0cate:WAVEform:FREE?</td>
<td>Return number of unallocated data points</td>
</tr>
<tr>
<td>ALL0cate:WAVEform:REF&lt;x&gt;?</td>
<td>Specify the number of allocated data points</td>
</tr>
<tr>
<td>DELETE:SETUp</td>
<td>Delete stored setup</td>
</tr>
<tr>
<td>DELETE:WAVEform</td>
<td>Delete stored waveform</td>
</tr>
<tr>
<td>*RCL</td>
<td>Recall settings</td>
</tr>
<tr>
<td>RECA11:SETUp</td>
<td>Recall saved instrument settings</td>
</tr>
<tr>
<td>RECA11:WAVEform (File System only)</td>
<td>Recall saved waveform</td>
</tr>
<tr>
<td>*SAV</td>
<td>Save settings</td>
</tr>
<tr>
<td>SAVE:SETUp</td>
<td>Save instrument settings</td>
</tr>
<tr>
<td>SAVE:WAVEform</td>
<td>Save waveform</td>
</tr>
<tr>
<td>SAVE:WAVEform:FILEFormat (Not on TDS 510A)</td>
<td>Specifies the file format for saved waveforms</td>
</tr>
</tbody>
</table>

**Status and Error Commands**

Table 2–18 lists the status and error commands the digitizing oscilloscope supports. These commands let you determine the status of the digitizing oscilloscope and control events.

Several commands and queries used with the digitizing oscilloscope are common to all devices on the GPIB bus. IEEE Std 488.2–1987 defines these commands and queries. They begin with an asterisk (*).

**Table 2–18: Status and Error Commands**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLEv?</td>
<td>Return all events</td>
</tr>
<tr>
<td>BUSY?</td>
<td>Return scope status</td>
</tr>
<tr>
<td>*CLS</td>
<td>Clear status</td>
</tr>
</tbody>
</table>
Table 2–18: Status and Error Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESE</td>
<td>Device event status enable</td>
</tr>
<tr>
<td>*ESE</td>
<td>Event status enable</td>
</tr>
<tr>
<td>*ESR?</td>
<td>Return standard event status register</td>
</tr>
<tr>
<td>EVENT?</td>
<td>Return event code</td>
</tr>
<tr>
<td>EVMsg?</td>
<td>Return event code and message</td>
</tr>
<tr>
<td>EVQty?</td>
<td>Return number of events in queue</td>
</tr>
<tr>
<td>ID?</td>
<td>Identification</td>
</tr>
<tr>
<td>*OPC</td>
<td>Operation complete</td>
</tr>
<tr>
<td>*OPT?</td>
<td>(Not in TDS 400A) Return installed options</td>
</tr>
<tr>
<td>*PSC</td>
<td>Power-on status clear</td>
</tr>
<tr>
<td>*PUD</td>
<td>Query or set User Protected Data</td>
</tr>
<tr>
<td>*RST</td>
<td>Reset</td>
</tr>
<tr>
<td>*SRE</td>
<td>Service request enable</td>
</tr>
<tr>
<td>*STB?</td>
<td>Read status byte</td>
</tr>
<tr>
<td>*WAI</td>
<td>Wait to continue</td>
</tr>
</tbody>
</table>

Trigger Commands

Trigger commands control all aspects of digitizing oscilloscope triggering. Table 2–19 lists these commands.

There are two triggers, main and delayed. Where appropriate, the command set has parallel constructions for each trigger.

You can set the main or delayed triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set TDS 510A, 500B, 600B, and 700A main triggers to pulse and logic modes. Pulse triggering lets the oscilloscope trigger whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The digitizing oscilloscope then triggers when it detects a certain combination of signal levels.
Table 2–19: Trigger Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER</td>
<td>Force trigger event; Return parameters</td>
</tr>
<tr>
<td>TRIGGER:DELay</td>
<td>Delay trigger level to 50%</td>
</tr>
<tr>
<td>TRIGGER:DELay:BY</td>
<td>Delay by time or events</td>
</tr>
<tr>
<td>TRIGGER:DELay:EDGE?</td>
<td>Return delay trigger parameters</td>
</tr>
<tr>
<td>TRIGGER:DELay:EDGE:COUPling</td>
<td>Delay trigger coupling</td>
</tr>
<tr>
<td>TRIGGER:DELay:EDGE:SLOpe</td>
<td>Delay trigger slope</td>
</tr>
<tr>
<td>TRIGGER:DELay:EDGE:SOURce</td>
<td>Delay trigger source</td>
</tr>
<tr>
<td>TRIGGER:DELay:EVENTS?</td>
<td>Return delay trigger event parameters</td>
</tr>
<tr>
<td>TRIGGER:DELay:EVENTS:COUn</td>
<td>Delay by events count</td>
</tr>
<tr>
<td>TRIGGER:DELay:LEVe1</td>
<td>Delay trigger level</td>
</tr>
<tr>
<td>TRIGGER:DELay:TIME</td>
<td>Time for delay by time</td>
</tr>
<tr>
<td>TRIGGER:DELay:TYPE</td>
<td>Delay trigger, edge</td>
</tr>
<tr>
<td>TRIGGER:MAIn</td>
<td>Main trigger level to 50%</td>
</tr>
<tr>
<td>TRIGGER:MAIn:EDGE?</td>
<td>Return main edge trigger parameters</td>
</tr>
<tr>
<td>TRIGGER:MAIn:EDGE:COUPling</td>
<td>Main trigger coupling</td>
</tr>
<tr>
<td>TRIGGER:MAIn:EDGE:SLOpe</td>
<td>Main trigger slope</td>
</tr>
<tr>
<td>TRIGGER:MAIn:EDGE:SOURce</td>
<td>Main trigger source</td>
</tr>
<tr>
<td>TRIGGER:MAIn:HOLDoff?</td>
<td>Return main trigger holdoff value</td>
</tr>
<tr>
<td>TRIGGER:MAIn:HOLDoff:ACTUal? (TDS 500B, 600B &amp; 700A)</td>
<td>Return main trigger holdoff value in seconds</td>
</tr>
<tr>
<td>TRIGGER:MAIn:HOLDoff:BY (TDS 500B, 600B &amp; 700A)</td>
<td>Main trigger holdoff default</td>
</tr>
<tr>
<td>TRIGGER:MAIn:HOLDoff:TIME (TDS 500B, 600B &amp; 700A)</td>
<td>Main trigger holdoff time</td>
</tr>
<tr>
<td>TRIGGER:MAIn:HOLDoff:VALue (TDS 400A &amp; 510A)</td>
<td>Main trigger holdoff value</td>
</tr>
<tr>
<td>TRIGGER:MAIn:LEVe1</td>
<td>Main trigger level</td>
</tr>
<tr>
<td>TRIGGER:MAIn:LOGIC? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Returns main logic trigger parameters</td>
</tr>
<tr>
<td>TRIGGER:MAIn:LOGIC:CLAss (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Logic trigger input usage</td>
</tr>
<tr>
<td>TRIGGER:MAIn:LOGIC:FUNCTION (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Logic trigger input combining</td>
</tr>
<tr>
<td>TRIGGER:MAIn:LOGIC:INPut? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Return main logic trigger input settings</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:INPUT:CH&lt;x&gt; (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Logic trigger expected channel state</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:PA Trotting:WHEn (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Main logic pattern trigger condition</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:PA Trotting:WHEn:LESSLImit (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Maximum time the selected pattern may be true and still generate main logic pattern trigger</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:PA Trotting:WHEn:MORELImit (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Minimum time the selected pattern may be true and still generate main logic pattern trigger</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: CLock:EDG (TDS 500B, 600B &amp; 700A)</td>
<td>Clock edge polarity for setup and hold violation triggering</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: CLock:LEVEL (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold clock voltage trigger level</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: CLock:SOURce (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold clock input source</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: DATa:LEVEL (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold data level</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: DATa:SOURce (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold data input data channel</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: HOLDTime (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold trigger hold time</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:SETHold: SETTime (TDS 500B, 600B &amp; 700A)</td>
<td>Setup/Hold trigger set time</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:STATE:INPUT:CH4 (TDS 510A, 500B 600B, &amp; 700A)</td>
<td>Logic trigger expected for channel 4 state class</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:STATE:WHEn (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>When the logic trigger occurs (on true or false)</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:THReshold? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Return main logic thresholds</td>
</tr>
</tbody>
</table>
Table 2–19: Trigger Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:MAIn:LOGic:THReshold: CH&lt;x&gt; (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Logic trigger thresholds</td>
</tr>
<tr>
<td>TRIGger:MAIn:LOGic:WHen (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Logic trigger on combination true or false</td>
</tr>
<tr>
<td>TRIGger:MAIn:MODE</td>
<td>Main trigger mode</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Returns pulse trigger parameters</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:CLAss (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger class</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:GLItch:FILTER (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Glitch filter on and off</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:GLItch:POLarity (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Glitch filter positive, negative, or both</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:GLItch:WIDth (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Glitch trigger with differentiation between glitch and valid pulse</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:RUNT? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Return runt trigger parameters</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:RUNT:POLarity (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Runt trigger positive, negative, or both</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:RUNT:THReshold:Both (TDS 500B, 600B &amp; 700A)</td>
<td>Trigger level switching thresholds</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:RUNT:WHen (TDS 500B, 600B &amp; 700A)</td>
<td>Runt pulse width type to check for</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:RUNT:WIDth (TDS 500B, 600B &amp; 700A)</td>
<td>Minimum width for valid main pulse runt trigger</td>
</tr>
<tr>
<td>TRIGger:MAIn:PULSe:SLEWRate:DELTAtime (TDS 500B, 600B &amp; 700A)</td>
<td>Slew rate trigger delta time</td>
</tr>
</tbody>
</table>
### Table 2–19: Trigger Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:POLarity (TDS 500B, 600B &amp; 700A)</td>
<td>Slew rate trigger polarity</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:SLEWRate? (TDS 500B, 600B &amp; 700A)</td>
<td>Return slew rate value</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:THReshold:BOTh (TDS 500B, 600B &amp; 700A)</td>
<td>Upper and lower slew rate trigger thresholds</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:THReshold:HIGH (TDS 500B, 600B &amp; 700A)</td>
<td>Upper limit for slew rate pulse</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:THReshold:LOW (TDS 500B, 600B &amp; 700A)</td>
<td>Lower limit for slew rate pulse</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SLEWRate:WHEn (TDS 500B, 600B &amp; 700A)</td>
<td>Slewing signal type to check for</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:SOURce (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger channel</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:TIMEOut:POLarity (TDS 500B, 600B, &amp; 700A)</td>
<td>Pulse timeout trigger polarity</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:TIMEOut:TIME (TDS 500B, 600B, &amp; 700A)</td>
<td>Pulse timeout trigger time</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:WIDTH? (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Return trigger pulse width parameters</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:WIDTH:HIGHLimit (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger maximum pulse width</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:WIDTH:LOWLimit (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger minimum pulse width</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:WIDTH:POLarity (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger positive, negative, or both</td>
</tr>
<tr>
<td>TRIGger:MAIn:PUlse:WIDTH:WHEn (TDS 510A, 500B, 600B, &amp; 700A)</td>
<td>Pulse trigger when pulse detected or when not detected</td>
</tr>
<tr>
<td>TRIGger:MAIn:TYPE</td>
<td>Set main trigger to edge, logic, pulse, or, with option 05, video type</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo? (Option 05)</td>
<td>Return video trigger parameters</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:BY (TDS 400A Option 05)</td>
<td>Set video trigger delay mode</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FIELD (Option 05)</td>
<td>Set video trigger field</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FIELDType (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set video trigger field type</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat? (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Return FlexFormat parameters</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat:FIELDRATE (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set FlexFormat frames per second</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat:FIELDS (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set FlexFormat video fields</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat: LINES (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set FlexFormat lines in a frame</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat:NEGSyncwidth (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set FlexFormat negative sync width</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat:VISTArttime (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set time from positive (+) edge of tri-sync pulse for the last line in the selected field to the leading edge (–) of the first negative vertical sync pulse</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:FLEXformat: VISTOptime (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set time from positive edge of tri-sync pulse for the last line in the selected field (t0) to trailing edge (positive) of the first negative vertical sync pulse</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: FLEXformat:V2STArttime (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set time from t0 to the leading edge (positive) of the second vertical sync pulse</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: FLEXformat:V2STOptime (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Set time from t0 to trailing edge (positive) of the second negative vertical sync pulse</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: HDTv (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Select high definition TV format</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: HOLdoff? (TDS 400A Option 05)</td>
<td>Return video trigger holdoff</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: HOLdoff:VALue (TDS 400A Option 05)</td>
<td>Set video trigger holdoff value</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo: INTERLace (TDS 400A Option 05)</td>
<td>Select video trigger interface format</td>
</tr>
<tr>
<td>TRIGger:MAIN:VIDeo:LINE (Option 05)</td>
<td>Set video trigger delay in terms of a number of lines</td>
</tr>
</tbody>
</table>
### Table 2–19: Trigger Commands (Cont.)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGger:MAIn:VIDeo:LINEs (TDS 400A Option 05)</td>
<td>Set video trigger delay in terms of a number of lines</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:NTSc (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Select color or mono NTSC</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:PAL (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Select color or mono PAL</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:SCAN (TDS 400A Option 05)</td>
<td>Set video trigger scan rate</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:SCANPeriod (TDS 400A Option 05)</td>
<td>Set video trigger scan period</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:SOURce Option 05</td>
<td>Select video trigger source</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:STANdard (TDS 510A, 500B, 600B, &amp; 700A Option 05)</td>
<td>Select video trigger standard</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:SYNC (Option 05)</td>
<td>Select video trigger sync polarity</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:SYStem (TDS 400A Option 05)</td>
<td>Select video trigger class</td>
</tr>
<tr>
<td>TRIGger:MAIn:VIDeo:TIME (TDS 400A Option 05)</td>
<td>Set video trigger delay time</td>
</tr>
<tr>
<td>TRIGger:STATE?</td>
<td>Return trigger system status</td>
</tr>
</tbody>
</table>
**Vertical Commands**

Vertical commands control the display of channels and of main and reference waveforms. Table 2–20 lists these commands.

The SElect:<wfm> command also selects the waveform many commands in other command groups use.

You may replace VOLts for SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix digitizing oscilloscopes.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH&lt;x&gt;?</td>
<td>Return vertical parameters</td>
</tr>
<tr>
<td>CH&lt;x&gt;:BANwidth</td>
<td>Channel bandwidth</td>
</tr>
<tr>
<td>CH&lt;x&gt;:COUpl ing</td>
<td>Channel coupling</td>
</tr>
<tr>
<td>CH&lt;x&gt;:DESkew (TDS 500B, 600B, &amp; 700A)</td>
<td>Channel deskew time</td>
</tr>
<tr>
<td>CH&lt;x&gt;:IMPedance</td>
<td>Channel impedance</td>
</tr>
<tr>
<td>CH&lt;x&gt;:OFFSet</td>
<td>Channel offset</td>
</tr>
<tr>
<td>CH&lt;x&gt;:POSition</td>
<td>Channel position</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBE?</td>
<td>Return channel probe attenuation</td>
</tr>
<tr>
<td>CH&lt;x&gt;:PROBECal? (TDS 500B, 600B, &amp; 700A)</td>
<td>Return channel probe cal status</td>
</tr>
<tr>
<td>CH&lt;x&gt;:SCAle</td>
<td>Channel volts per div</td>
</tr>
<tr>
<td>CH&lt;x&gt;:VOLts</td>
<td>Same as CH&lt;x&gt;:SCAle</td>
</tr>
<tr>
<td>MATH&lt;x&gt;?</td>
<td>Return math waveform definition</td>
</tr>
<tr>
<td>MATH&lt;x&gt;:DEF ine</td>
<td>Define math waveform</td>
</tr>
<tr>
<td>MATH&lt;x&gt;:NUMAVg (TDS 510A, 500B, 600B, &amp; 700A, some models require Option 2F)</td>
<td>Acquisition number at which to begin exponential averaging</td>
</tr>
<tr>
<td>MATH&lt;x&gt;:PROCessing (TDS 510A, 500B, 600B, &amp; 700A, some models require Option 2F)</td>
<td>Math waveform averaging on or off</td>
</tr>
<tr>
<td>SELe ct?</td>
<td>Return selected waveform</td>
</tr>
<tr>
<td>SELe ct:CONTRO1</td>
<td>Front-panel channel selector</td>
</tr>
<tr>
<td>SELe ct:&lt;wfm&gt;</td>
<td>Set selected waveform</td>
</tr>
</tbody>
</table>
Waveform Commands

Waveform commands let you transfer waveform data points to and from the digitizing 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 enveloped waveforms, each data value is either the min or max of a min/max pair. Before you transfer waveform data, you must specify the data format, record length, and waveform locations.

Table 2–21 lists these commands.

Waveform Data Formats

Acquired waveform data uses either one or two 8-bit data bytes to represent each data point. The number of bytes used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple, ENVelope, or PEAKde-tect mode uses one 8-bit byte per waveform data point. Data acquired in HIRes or AVerage mode uses two 8-bit bytes per point. For more information on the acquisition modes see the ACQuire: MODe command on page 2–40.

The DATa:WIDth command lets you specify the number of bytes per data point when transferring data to and from the digitizing oscilloscope. If you specify two bytes for data that uses only one, the least significant byte will be filled with zeros. If you specify one byte for data that uses two, the least significant byte will be ignored.

The digitizing oscilloscope can transfer waveform data in either ASCII or binary format. You specify the format with the DATa:ENCdg command.

**ASCII data** — is represented by signed integer values. The range of the 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 to five characters to represent the value, another character, if the value is negative, to represent a minus sign, and a comma to separate the data points.

An example ASCII waveform data string may look like this:

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

Use ASCII to obtain more human readable and easier to format output than binary. However, it may require more bytes to send the same values with ASCII than it does with binary. This may reduce transmission speeds.

**Binary data** — can be represented by signed integer or positive integer values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from –128 to 127, and positive integer
values range from 0 to 255. When the byte width is two, the values range from –32768 to 32767.

The defined binary formats also specify the order in which the bytes are transferred. The four binary formats are 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 Lengths

You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. 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 digitizing oscilloscope, you must specify the location of the first data point within the waveform record. For example, when you set DATa:START to 1, data points will be stored starting with the first point in the record, and when you set DATa:START to 500, data will be stored starting at the 500th point in the record. The digitizing oscilloscope will ignore DATa:STOP when reading in data as it will stop reading data when it has no more data to read or when it has reached the specified record length.

When transferring data from the digitizing oscilloscope, you must specify the first and last data points in the waveform record. Setting DATa:START to 1 and DATa:STOP to the record length will always return the entire waveform. You can also use the vertical bar cursors to delimit the portion of the waveform that you want to transfer. DATa:START and DATa:STOP can then be set to the current cursor positions by sending the command DATa SNAP.

Waveform Data Locations and Memory Allocation

The DATa:SOURce command specifies the data location when transferring waveforms from the digitizing oscilloscope. You can transfer out multiple waveforms at one time by specifying more than one source.

You can transfer in to the digitizing oscilloscope only one waveform at a time. Waveforms sent to the oscilloscope are always stored in one of the four reference memory locations. You can specify the reference memory location with the DATa:DESTination command. You must define the memory size for the specified location before you store the data. The ALLOCate: WAVEFORM:REF<x> command lets you specify the memory size for each reference location.
**Waveform Preamble**

Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in place when the waveform was created. Refer to the WFMPre commands starting on page 2–291 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. The GETWM program on the diskettes that come with this manual shows how you can scale data.

**Transferring Waveform Data from the TDS Family Oscilloscope**

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

1. Select the waveform source(s) using the DATa:SOUrce command. If you want to transfer multiple waveforms, select more than one source.
2. Specify the waveform data format using DATa:ENCdg.
3. Specify the number of bytes per data point using DATa:WIDth.
4. Specify the portion of the waveform that you want to transfer using DATa:STARt and DATa:STOP.
5. Transfer waveform preamble information using WFMPre? query.
6. Transfer waveform data from the digitizing oscilloscope using the CURVe? query.

**Transferring Waveform Data to the TDS Family Oscilloscope**

You can transfer waveform data to one of the four reference memory locations in the digitizing oscilloscope using the following sequence:

2. Specify the memory size for the reference location specified in Step 1 using the ALLocate:WAVEFORM:REF<x> command.
3. Specify the waveform data format using DATa:ENCdg.
4. Specify the number of bytes per data point using DATa:WIDth.
5. Specify first data point in the waveform record using DATa:STARt.
6. Transfer waveform preamble information using WFMPre:<wfm>.
7. Transfer waveform data to the digitizing oscilloscope using CURVe.
Table 2–21: Waveform Commands

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURve</td>
<td>Transfer waveform data</td>
</tr>
<tr>
<td>DATa</td>
<td>Waveform data format and location</td>
</tr>
<tr>
<td>DATa:DESTination</td>
<td>Destination for waveforms sent to digitizing oscilloscope</td>
</tr>
<tr>
<td>DATa:ENCdg</td>
<td>Waveform data encoding method</td>
</tr>
<tr>
<td>DATa:SOURce</td>
<td>Source of CURve? data</td>
</tr>
<tr>
<td>DATa:START</td>
<td>Starting point in waveform transfer</td>
</tr>
<tr>
<td>DATa:STOP</td>
<td>Ending point in waveform transfer</td>
</tr>
<tr>
<td>DATa:TARGet</td>
<td>Same as DATa:DESTination</td>
</tr>
<tr>
<td>DATa:WIDth</td>
<td>Byte width of waveform points</td>
</tr>
<tr>
<td>WAVFrm?</td>
<td>Return waveform preamble and data</td>
</tr>
<tr>
<td>WAVPre?</td>
<td>Return waveform format data</td>
</tr>
<tr>
<td>WFMPre:BIT_Nr</td>
<td>Preamble bit width of waveform points.</td>
</tr>
<tr>
<td>WFMPre:BN_FMT</td>
<td>Preamble binary encoding type</td>
</tr>
<tr>
<td>WFMPre:BYT_Nr</td>
<td>Preamble byte width of waveform points</td>
</tr>
<tr>
<td>WFMPre:BYT_Or</td>
<td>Preamble byte order of waveform points</td>
</tr>
<tr>
<td>WFMPre:CRVchk</td>
<td>Preamble checksum of waveform points</td>
</tr>
<tr>
<td>WFMPre:ENCdg</td>
<td>Preamble encoding method</td>
</tr>
<tr>
<td>WFMPre:NR_Pt</td>
<td>Number of points in the curve</td>
</tr>
<tr>
<td>WFMPre:PT_FMT</td>
<td>Format of curve points</td>
</tr>
<tr>
<td>WFMPre:PT_Offset</td>
<td>Trigger position</td>
</tr>
<tr>
<td>WFMPre:WFId</td>
<td>Curve identifier</td>
</tr>
<tr>
<td>WFMPre:XINcr</td>
<td>Horizontal sampling interval</td>
</tr>
<tr>
<td>WFMPre:XMULT</td>
<td>Horizontal scale factor</td>
</tr>
<tr>
<td>WFMPre:XOFF</td>
<td>Horizontal offset</td>
</tr>
<tr>
<td>WFMPre:XUNIT</td>
<td>Horizontal units</td>
</tr>
<tr>
<td>WFMPre:ZERO</td>
<td>Horizontal origin offset</td>
</tr>
<tr>
<td>WFMPre:YMULT</td>
<td>Vertical scale factor</td>
</tr>
<tr>
<td>WFMPre:YOFF</td>
<td>Vertical offset</td>
</tr>
<tr>
<td>WFMPre:YUNIT</td>
<td>Vertical units</td>
</tr>
<tr>
<td>WFMPre:ZERO</td>
<td>Offset voltage</td>
</tr>
<tr>
<td>WFMPre:ZMULT</td>
<td>Z-axis scale factor</td>
</tr>
<tr>
<td>WFMPre:ZOFF</td>
<td>Z-axis offset</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>WFMPre:ZUnit</code></td>
<td>Z-axis units</td>
</tr>
<tr>
<td><code>WFMPre:ZZero</code></td>
<td>Z-axis origin offset</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:NR_Pt</code></td>
<td>Number of points in the curve</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:PT_Fmt</code></td>
<td>Format of curve points</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:PT_Off</code></td>
<td>Trigger position</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:WId</code></td>
<td>Curve identifier</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:XINcr</code></td>
<td>Horizontal sampling interval</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:XUnit</code></td>
<td>Horizontal units</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:XZErO</code> (TDS 500B, 600B, &amp; 700A)</td>
<td>Horizontal units</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:YMUlt</code></td>
<td>Vertical scale factor</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:YOFF</code></td>
<td>Vertical offset</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:YUnit</code></td>
<td>Vertical units</td>
</tr>
<tr>
<td><code>WFMPre:&lt;wfm&gt;:YZErO</code></td>
<td>Offset voltage</td>
</tr>
</tbody>
</table>
Zoom Commands

Zoom commands let you expand and position the waveform display horizontally and vertically without changing the time base or vertical settings. Table 2–22 lists these commands.

**Table 2–22: Zoom Commands**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z00m</td>
<td>Reset zoom parameters to defaults</td>
</tr>
<tr>
<td>Z00m:DUAL (TDS 400A, 500B, 600B, &amp; 700A)</td>
<td>Turn dual zoom mode on and off</td>
</tr>
<tr>
<td>Z00m:DUAL:OFFSET (TDS 400A, 500B, 600B, &amp; 700A)</td>
<td>Adjust the requested horizontal offset between the centers of the main and second zoom boxes</td>
</tr>
<tr>
<td>Z00m:GRAticle (TDS 400A, 500B, 600B, &amp; 700A)</td>
<td>Select between the upper and lower graticule for use by the zoom preview state</td>
</tr>
<tr>
<td>Z00m:HORIZONTAL:LOCK</td>
<td>Horizontal zoom lock</td>
</tr>
<tr>
<td>Z00m:HORIZONTAL:POSITION</td>
<td>Horizontal zoom position</td>
</tr>
<tr>
<td>Z00m:HORIZONTAL:SCALE</td>
<td>Horizontal zoom scale</td>
</tr>
<tr>
<td>Z00m:STATE</td>
<td>Turn zoom mode on or off</td>
</tr>
<tr>
<td>Z00m:VERTICAL:POSITION</td>
<td>Vertical zoom position</td>
</tr>
<tr>
<td>Z00m:VERTICAL:SCALE</td>
<td>Vertical zoom scale</td>
</tr>
</tbody>
</table>
You can use commands to either set instrument features or query instrument values. You can use some commands to do both, some to only set, and some to only query. This manual marks set only commands with the words “No Query Form” included with the command name. It marks query only commands with a question mark appended to the header, and includes the words “Query Only” in the command name.

This manual 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.

**ACQuire? (Query Only)**

Returns all the current acquisition parameters.

**Group**

Acquisition

**Syntax**

ACQuire?

**Examples**

ACQUIRE?
might return the string:
ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1 for the current acquisition parameters.

**ACQuire:AUTOSAve**

*TDS 500B, 600B & 700A Only*

Saves waveforms in reference memory when acquisition completes. This is equivalent to setting Autosave Single Seq in the Acquire menu and the corresponding side menu Off or On items.

When you start a Single Sequence with Autosave set to ON, the oscilloscope nulls out all existing reference waveforms. At the end of Single Sequence, the oscilloscope saves all displayed live channels to reference waveform memory. It saves references in the order Ch1 -> Ref1, Ch2 -> Ref2, Ch3 -> Ref3, Ch4 -> Ref4. The exact number of references saved may depend on the record length used.
Command Descriptions

**Group**  
Acquisition

**Syntax**  
ACQuire:AUTOSAve { OFF | ON | <NR1> }

ACQuire:AUTOSAve?

**Arguments**  
OFF or <NR1> = 0 turns autosave off.

ON or <NR1> ≠ 0 turns autosave on.

**Examples**  
ACQUIRE:AUTOSAVE 1  
turns autosave mode on.

ACQUIRE:AUTOSAVE OFF  
turns autosave mode off.

ACQUIRE:AUTOSAVE?  
might return 1, indicating that autosave mode is on.

**ACQuire:MODe**

Sets or queries the acquisition mode of the digitizing oscilloscope. 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 set by the horizontal scale (time per division). The digitizing oscilloscope sampling system always samples the digitizing oscilloscope at the maximum rate, and so an acquisition interval may include more than one sample.

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

**Group**  
Acquisition
Related Commands

ACQuire:NUMA Vg, ACQuire:NUMENv, CURVe?, DATa:WIDth

Syntax

For the TDS 400A, 510A, 500B & 700A:

\[
\text{ACQuire:MODe} \{ \text{SAMPle} | \text{PEAKdetect} | \text{HIREs} | \text{AVERAGE} | \text{ENVelope} \}
\]

For the TDS 600B:

\[
\text{ACQuire:MODe} \{ \text{SAMPle} | \text{PEAKdetect} | \text{AVERAGE} | \text{ENVelope} \}
\]

For all TDS:

\[
\text{ACQuire:MODe}?
\]

For the TDS 400A, 510A, 500B, & 700A:

For the TDS 600B:

Arguments

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

PEAKdetect specifies the display of the high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. PEAKdetect mode can reveal the presence of aliasing or narrow spikes.
HIRes (for the TDS 400A, 510A, 500B, & 700A) specifies Hi Res mode, where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

AVERAGE specifies averaging mode, 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.

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

**Examples**

```
ACQUIRE:MODE ENVELOPE
```

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

```
ACQUIRE:MODE?
```

might return ENVELOPE.

**ACQUIRE:NUMACq? (Query Only)**

Indicates the number of acquisitions that have taken place since starting acquisition. This value is reset to zero when any Acquisition, Horizontal, or Vertical arguments that affect the waveform are modified. The maximum number of acquisitions that can be counted is 2\(^{30}\) – 1. Counting stops when this number is reached. This is the same value that is displayed in the upper center of the screen when the acquisition system is stopped.
Examples

ACQUIRE:NUMACQ?
might return 350, indicating that 350 acquisitions took place since an AC-
QUIRE:STATE RUN command was executed.

ACQuire:NUMAVg

Sets the number of waveform acquisitions that make up an averaged waveform. This is equivalent to setting the Average count in the Acquisition Mode side menu.

Group

Acquisition

Related Commands

ACQuire:MODe

Syntax

ACQuire:NUMAVg <NR1>

ACQuire:NUMAVg?

Arguments

<NR1> is the number of waveform acquisitions, from 2 to 10,000.

Examples

ACQUIRE:NUMAVG 10

specifies that an averaged waveform will show the result of combining 10 separately acquired waveforms.

ACQUIRE:NUMAVG?

might return 75, indicating that there are 75 acquisitions specified for averaging.

ACQuire:NUMEnv

Sets the number of waveform acquisitions that make up an envelope waveform. This is equivalent to setting the Envelope count in the Acquisition Mode side menu.

Group

Acquisition

Related Commands

ACQuire:MODe
Syntax

\[ \text{ACQuire:NUMEnv} \{ \text{<NR1> | INFInite} \} \]

\[ \text{ACQuire:NUMEnv?} \]

Arguments

\(<\text{NR1}> \neq 0\) is the number of waveform acquisitions, from 1 to 2000. The envelope will restart after the specified number of envelopes have been acquired or when the ACQuire:STATE RUN command is sent.

\(\text{INFI}n\text{t}\) or \(<\text{NR1}> = 0\) specifies continuous enveloping.

**NOTE**

If you set the acquisition system to single sequence, envelope mode, and set the number of envelopes to infinity, the digitizing oscilloscope will envelope a maximum of 2001 acquisitions.

Examples

\[ \text{ACQUIRE:NUMENV 10} \]

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

\[ \text{ACQUIRE:NUMENV?} \]

might return 0, indicating that acquisitions are acquired infinitely for enveloped waveforms.

**ACQuire:REPEt**

*TD 400A, 510A, 500B, & 700A Only*

Controls repetitive signal acquisition. This is equivalent to setting **Repetitive Signal** in the Acquire menu. When the digitizing oscilloscope is in real-time operation, this setting has no effect.

The ACQuire:REPEt command specifies the behavior of the acquisition system during equivalent-time (ET) operation. When repetitive mode is on, the acquisition system will continue to acquire waveform data until the waveform record is filled with acquired data. When repetitive mode is off and you specify single acquisition operation, only some of the waveform data points will be set with acquired data, and the displayed waveform shows interpolated values for the unsampled data points.
**ACQquire:REPEt**

**Group**  
Acquisition

**Related Commands**  
ACQquire:STATE, ACQquire:STOPAfter

**Syntax**  
ACQquire:REPEt { OFF | ON | <NR1> }

ACQquire:REPEt?

**Arguments**  
OFF or <NR1> = 0 turns repetitive mode off.

ON or <NR1> ≠ 0 turns repetitive mode on.

**Examples**  
ACQquire:REPEt 1  
turns repetitive mode on.

ACQquire:REPEt OFF  
turns repetitive mode off.

ACQquire:REPEt?  
might return 1, indicating that repetitive signal acquisition mode is on.

**ACQquire:STATE**

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

**Group**  
Acquisition

**Related Commands**  
ACQquire:NUMACq?, ACQquire:REPEt, ACQquire:STOPAfter

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

ACQquire:STATE?
**ACQuire:STOPAfter**

Tells the digitizing oscilloscope when to stop taking acquisitions. This is equivalent to setting **Stop After** in the Acquire menu.

**Arguments**

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

ON or RUN or <NR1> ≠ 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 prior to 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.

**Group**

Acquisition

**Related Commands**

ACQuire:MODe, ACQuire:STATE, ACQuire:REPEt

**Syntax**

ACQuire:STOPAfter { RUNStop | SEQuence | LIMit }

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 digitizing 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, and the horizontal scale is set to a speed that allows real-time operation, then the digitizing oscilloscope will stop after digitizing a waveform from a single trigger event. However, if the acquisition mode is set to average 100 waveforms, then the digitizing oscilloscope will stop only after all 100 waveforms have been acquired. The ACQuire: STATE command and the front-panel **RUN/STOP** button will also stop acquisition when the digitizing oscilloscope is in single sequence mode.

LIMIT specifies the digitizing oscilloscope stops after the limit test condition is met.

**NOTE** If you set the acquisition system to single sequence, envelope mode, and set the number of envelopes to infinity, the digitizing oscilloscope will envelope a maximum of 2001 acquisitions.

**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.

**ALIas**

Turns command aliases on or off. This command is identical to the ALIAs:STATE command.

**Group**

Alias

**Syntax**

ALIas { OFF | ON | <NR1> }

ALIas?
Arguments

OFF or <NR1> = 0 turns alias expansion off. If a defined alias label is sent when ALIAS is OFF, an execution error (110, “Command header error”) will be generated.

ON or <NR1> ≠ 0 turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples

ALIAS ON
turns the alias feature on.

ALIAS?
returns 1 when aliases are on.

ALIAS:CATalog? (Query Only)

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

Group
Alias

Syntax
ALIAS:CATalog?

Returns
<QString>[,<QString>...]

Examples
ALIAS:CATalog?
might return the string "SETUP1","TESTMENU1","DEFAULT", showing there are 3 aliases named SETUP1, TESTMENU1, and DEFAULT.

ALIAS:DEFIne

Assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query.
provided ALIas:STATE has been turned ON. The ALIas:DEFIne? query returns the definition of a selected alias.

Up to 10 aliases can be defined at one time. Aliases can be recursive. That is, aliases can include other aliases with up to 10 levels of recursion.

**Group**  Alias

**Syntax**  ALIas:DEFIne <QString><Comma>{ <QString> | <Block> }  
ALIas:DEFIne? <QString>

**Arguments**  
The first <QString> is the alias label. This label cannot be a command name. Labels must start with a letter, and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be ≤12 characters.

The second <QString> or <Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands (see page 2–4). The sequence must be ≤80 characters.

---

**NOTE**  Attempting to give two aliases the same name causes an execution error. To give a new alias the name of an existing alias, you must first delete the existing alias.

---

**Examples**  
ALIAS:DEFINE "ST1",";RECALL:SETUP 5;AUTOSET EXECUTE;:SELECT:CH1 ON"
defines an alias named “ST1” that sets up the digitizing oscilloscope.

ALIAS:DEFIne? "ST1"
might return :ALIAS:DEFIne "ST1",#239:RECALL:SETUP 5;AUTOSET EXECUTE;:SELECT:CH1 ON
**ALIas:DELEte (No Query Form)**

Removes a specified alias. This command is identical to ALIas:DELEte:NAME.

**Group**
Alias

**Syntax**
`ALIas:DELEte <QString>`

**Arguments**
`<QString>` is the name of the alias you want to remove. Using ALIas:DELEte without specifying an alias causes an execution error. `<QString>` must be a previously defined alias.

**Examples**
`ALIAS:DELETE "SETUP1"`
deletes the alias named SETUP1.

**ALIas:DELEte:ALL (No Query Form)**

Deletes all existing aliases.

**Group**
Alias

**Syntax**
`ALIas:DELEte:ALL`

**Examples**
`ALIAS:DELETE:ALL`
deletes all aliases.

**ALIas:DELEte:NAME (No Query Form)**

Removes a specified alias. This command is identical to ALIas:DELEte.

**Group**
Alias

**Syntax**
`ALIas:DELEte:NAME <QString>`
Arguments  

<QString> is the name of the alias to remove. Using ALIas:DELETE:NAME without specifying an alias causes an execution error. <QString> must be a previously defined alias.

Examples  

ALIAS:DELETE:NAME "STARTUP"
deletes the alias named STARTUP.

ALIas:STATE

Turns aliases on or off. This command is identical to the ALIas command.

Group  

Alias

Syntax  

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

ALIas:STATE?

Arguments  

OFF or <NR1> = 0 turns alias expansion off. If a defined alias is sent when ALIas:STATE is OFF, a command error (102) will be generated.  

ON or <NR1> ≠ 0 turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples  

ALIAS:STATE OFF  
turns the command alias feature off.  

ALIAS:STATE?  
returns 0 when alias mode is off.

ALLEv? (Query Only)

Causes the digitizing 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 digitizing oscilloscope. As much of the command will be 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; ".

**ALLOcate? (Query Only)**

Returns the number of data points allocated for all four reference memory locations.

**Group** Save and Recall

**Syntax** ALLOcate?
Examples

ALLOCATE?
might return :ALLOCATE:WAVEFORM:REF1 50000;REF2 0;REF3 0; REF4 0;, indicating that all 50000 data points are allocated to reference memory location 1.

ALLOCATE:WAVEform? (Query Only)

Returns the number of data points allocated for all four reference memory locations.

Group
Save and Recall

Syntax
ALLOCATE:WAVEform?

Examples

ALLOCATE?
might return :ALLOCATE:WAVEFORM:REF1 500;REF2 500;REF3 500; REF4 0;, indicating that 500 data points are allocated to each of the first three reference memory locations.

ALLOCATE:WAVEform:FREE? (Query Only)

Returns the approximate number of data points that have not been allocated.

Group
Save and Recall

Syntax
ALLOCATE:WAVEform:FREE?

Returns
<NR1> is the approximate number of data points available.

Examples

ALLOCATE:WAVEFORM:FREE?
might return 520 indicating that there are approximately 500 data points available for allocation. The extra 20 are used for administration purposes.
**ALLOcate:WAVEform:REF<x>**

Sets or queries the number of waveform data points for the specified reference location. If an attempt is made to allocate memory when it is not available, an execution error is generated and the memory is not allocated.

**Group**  
Save and Recall

**Syntax**  
ALLOcate:WAVEform:REF<x>  
ALLOcate:WAVEform:REF<x>?  

**Arguments**  

\(<\text{NR}\text{1}> = 0 \) is returned when the reference location is empty.

\(<\text{NR}\text{1}> \neq 0 \) specifies the number of data points. Table 2–23 shows the number of data points supported for reference locations by TDS model. In the TDS 500B and 600B all invalid values less than the maximum will be forced to the next highest valid value, and those higher than the maximum will be forced to the maximum. For example, 15002 points on a TDS 520B with option 1M will allocate 50000 points of data for the reference. No complete references are stored for 500000 data points in the TDS 700A.

**Table 2–23: Waveform Data Points Supported for Reference Locations**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>500, 1000, 2500, 5000, 15000, 30000 (4 refs)</td>
<td>500, 1000, 2500, 5000, 15000, 50000</td>
<td>500, 1000, 2500, 5000, 15000, 50000</td>
<td>500, 1000, 2500, 5000, 15000, 50000 (4 refs)</td>
</tr>
<tr>
<td>Option 1M (not on the TDS 510A)</td>
<td>60000 (2 refs), 120000 (1 ref)</td>
<td>TDS 520B, TDS 724A, TDS 782A only: 75000 (3 refs), 100000 (2 refs), 130000 (2 refs), 250000 (1 ref)</td>
<td>75000 (3 refs), 100000 (2 refs), 130000 (2 refs), 250000 (1 ref)</td>
<td>500000 (no complete refs)</td>
</tr>
</tbody>
</table>
Examples

ALLOCATE:WAVEFORM:REF2 1000
reserves 1,000 data points for REF2.

ALLOCATE:WAVEFORM:REF1?
might return 500

APPMenu

Displays the user-definable Application menu, and the query returns the current Application menu labels and title. This is equivalent to pressing the front-panel APPLICATION button.

Group

Application Menu

Related Commands

CLEARMenu, *ESR, EVENT?

Syntax

APPMenu ACTivate

APPMenu?

Arguments

ACTivate displays the Application menu. Use the CLEARMenu command to deactivate the Application menu.

Once the Application menu is activated, whenever a front-panel menu button is pressed an event is generated that tells which button was pressed. See page 3–17 for event codes.

Menu button presses will also generate Service Requests when the URQ bit is enabled in DESER and ESER and the ESB bit is enabled in SRER. See page 3–1 for a complete discussion of the use of these registers.

Examples

APPMENU ACTIVATE
displays the application menu.
**APPMenu:LABel**

Removes all user-defined Application menu button labels from the display. The APPMenu:LABel? query returns all the current label settings.

**Group**  
Application Menu

**Syntax**  
APPMenu:LABel CLEar  
APPMenu:LABel?

**Arguments**  
CLEar removes the main and side menu button labels from the display. Front-panel bezel button presses will continue to generate events.

**Examples**  
APPMENU:LABEL CLEAR  
clears the user-defined menu labels from the display.

**APPMenu:LABel:BOTTOM<x>**

Defines a label for the main menu button that is specified by <x>. Main menu buttons are located along the bottom of the display, and are numbered from 1 to 7 starting with the left-most button.

**Group**  
Application Menu

**Syntax**  
APPMenu:LABel:BOTTOM<x> <QString>  
APPMenu:LABel:BOTTOM<x>?

**Arguments**  
<QString> is the menu button label and can include any of the characters shown in the TDS Character Chart in Appendix A. The maximum length of the label is 1000 characters. The TDS displays the label in the area above the specified main menu button.
The TDS displays the label on a single line and centers it, both vertically and horizontally, within the label area. You can embed a line feed character in the string to position the label on multiple lines. You can also use white space tab characters to position the label within a line.

You can send a tab by transmitting a tab character (decimal 9) followed by two characters representing the most significant eight bits followed by the least significant eight bits of a 16-bit number. The number specifies the pixel column relative to the left margin of the label area. For example, to tab to pixel 13, send TAB (decimal 9), NUL (decimal 0), and CR (decimal 13).

The ESC @ character turns reverse video on and off, and can be embedded in the label string. The first ESC @ character displays all text following the ESC @ in reverse video until another ESC @ character is found in the string.

**NOTE.** The use of any undocumented codes may produce unpredictable results.

The label area is 45 pixels high and 90 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies. The label area is about 10 characters wide and 3 lines high. For a complete list of character widths in pixels, see Table A–1 on page A–1.

If the label exceeds the limits of the label area, either horizontally or vertically, the portion of the label that exceeds the limits will not be displayed. The label itself is not altered. The entire label can be returned as a query response regardless of what is displayed.

**Examples**

APPMenu:LABEL:BOTTOM3 "SETUP1"

assigns the label “SETUP1” to the third main menu button.

**APPMenu:LABel:RIGHT<x>**

Defines a label for the side menu button that is specified by <x>. Side menu buttons are located on the right side of the display, and are numbered from 1 to 5 starting with the top-most button.

**Group**  Application Menu

**Syntax**

APPMenu:LABel:RIGHT<x> <Qstring>

APPMenu:LABel:RIGHT<x>?
is the menu button label and can include any of the characters shown in the TDS Character Chart in Appendix A. The maximum length of the label is 1000 characters. The label is displayed in the area to the left of the specified side menu button. Refer to the APPMenu:LABel:BOTTOM<x> command on page 2–57 for more information on defining menu labels.

The label area is 72 pixels high and 112 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies. The label area is about 12 characters wide and 4 lines high. For a complete list of character widths in pixels, see Table A–1 on page A–1.

**Examples**

```plaintext
APPMenu:LABEL:RIGHT1 "TEST ON"
```

displays the label “TEST ON” next to the top side menu button.

### APPMenu:TITLE

Sets or queries the user-defined application menu title. The title is displayed above the side menu.

**Group**

Application Menu

**Related Commands**

APPMenu, APPMenu:LABel

**Syntax**

```
APPMenu:TITLE <QString>
```

```
APPMenu:TITLE?
```

**Arguments**

<QString> is the side menu title and can include any of the characters shown in the TDS Character Chart in Appendix A. The maximum length of the title is 1000 characters. The APPMenu:LABel:BOTTOM<x> command on page 2–57 provides information on defining menu labels.

The label area is 40 pixels high and 112 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies.
characters varies. The label area is about 12 characters wide and 4 lines high. For a complete list of character widths in pixels, see Table A–1 on page A–1.

**Examples**

APPMENU:TITLE "Custom Menu"
displays the title “Custom Menu” on the screen.

APPMENU:TITLE?
might return "Test Setup" for the current application menu title.

**AUTOSet (No Query Form)**

Causes the digitizing oscilloscope to adjust its vertical, horizontal, and trigger controls to provide a stable display of the selected waveform. This is equivalent to pressing the front-panel **AUTOSet** button. For a detailed description of the autoset function, see Autoset in the Reference section of the User Manual for your instrument.

**Group**

Miscellaneous

**Syntax**

AUTOSet EXECute

**Arguments**

EXECute autosets the displayed waveform.

**BELl (No Query Form)**

Beeps the audio indicator of the digitizing oscilloscope.

**Group**

Miscellaneous

**Syntax**

BELl

**Examples**

BELl
rings the bell.
**BUSY? (Query Only)**

Returns the status of the digitizing oscilloscope. This command allows you to synchronize the operation of the digitizing oscilloscope with your application program. Synchronization methods are described on page 3–7.

**Group**  
Status and error

**Related Commands**  
*OPC, *WAI

**Syntax**  
BUSY?

**Returns**  
<NR1> = 0 means that the digitizing oscilloscope is not busy processing a command whose execution time is extensive. These commands are listed in Table 2–24.

<NR1> = 1 means that the digitizing oscilloscope is busy processing one of the commands listed in Table 2–24.

**Table 2–24: Commands that Affect BUSY? Response**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single sequence acquisition</td>
<td>ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQuence)</td>
</tr>
<tr>
<td>Hardcopy output</td>
<td>HARDCopy START</td>
</tr>
</tbody>
</table>

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

**CAL? (Query Only)**

Instructs the digitizing oscilloscope to perform an internal self-calibration and return its calibration status.

**NOTE.** The self-calibration can take several minutes to respond. No other commands will be executed until calibration is complete.
**Group** Calibration and Diagnostic

**Syntax** *CAL?*

![Diagram](image)

**Returns**
- \(<\text{NR1}\> = 0\) indicates that the calibration completed without any errors detected.
- \(<\text{NR1}\> \neq 0\) indicates that the calibration did not complete successfully.

**Examples**
- *CAL?* performs an internal self-calibration and might return 0 to indicate that the calibration was successful.

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

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

**Group** Vertical

**Syntax** CH<x>?

![Diagram](image)

**Examples**
- CH1? might return the string :CH1:SCALE 10.0E-3;POSITION 0.0E+0;
  OFFSET 0.0E+0;COUPLING DC;IMPEDANCE MEG;BANDWIDTH FULL for channel 1.

### CH<x>:BANwidth

Sets or queries the bandwidth setting of the specified channel. This is equivalent to setting Bandwidth in the Vertical menu.

**Group** Vertical

**Syntax** CH<x>:BANwidth \{ TWEnty | HUNDred (TDS 400A, 510A) | TWOfifty (TDS 500B, 600B & 700A) | FULL \}
CH<x>:BANDwidth?

Arguments

Twenty sets the channel bandwidth to 20 MHz.

HUNDred sets the channel bandwidth to 100 MHz (TDS 400A, 510A).

TWOifty sets the channel bandwidth to 250 MHz (TDS 500B, 600B, & 700A).

FULL sets the channel bandwidth to the full bandwidth of the digitizing oscilloscope.

Examples

CH2:BANDWIDTH TWENTY
sets the bandwidth of channel 2 to 20 MHz.

CH1:BANDWIDTH?
might return FULL, which indicates that there is no bandwidth limiting on channel 1.

CH<x>:COUPling

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

Group

Vertical

Related Commands

CH<x>:IMPedance

Syntax

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

CH<x>:COUPling?
**CH<x>:COUPLING**

Sets or queries the coupling for this channel.

**Arguments**

- AC sets the specified channel to AC coupling.
- DC sets the specified channel to DC coupling.
- GND sets the specified channel to ground. Only a flat ground-level waveform will be displayed.

**Examples**

- CH1:COUPLING AC establishes AC coupling on channel 1.
- CH3:COUPLING? might return DC, indicating that channel 3 is set to DC coupling.

**CH<x>:DESKew**

*TDS 500B, 600B, & 700A Only*

Sets or queries the deskew time for this channel. This is equivalent to setting Deskew in the Vertical menu and entering a value with the keypad or general purpose knob.

Deskew allows you to compensate for time delay differences caused by signals coming in from cables of different length.

**Arguments**

- <NR3> is the deskew time for this channel. The range is –25.0 ns to +25.0 ns with a resolution of 1 ps. Out of range values are clipped.
**CH<x>:IMPedance**

Sets or queries the impedance setting at the specified input channel. This is equivalent to setting **Impedance** in the Ch<x> Coupling Impedance side menu.

TDS 500B, 600B and 700A: When you attach an active 50 Ω probe to an input channel of the TDS 500B, 600B or 700A, the oscilloscope reduces the maximum vertical scale from 10 V to 1 V per division. For example, an active 10X probe would provide 10 V per division and a passive 10X probe would provide 100 V per division.

**Group**
Vertical

**Related Commands**
CH<x>:COUPling

**Syntax**

```
CH<x>:IMPedance { FIFty | MEG }
CH<x>:IMPedance?
```

**Arguments**

- **FIFty** sets the specified channel to 50 Ω impedance.
- **MEG** sets the specified channel to 1 MΩ impedance.

**Examples**

```
CH1:IMPEDANCE FIFty
```

establishes 50 Ω impedance on channel 1.

```
CH3:IMPEDANCE?
```

might return MEG, indicating that channel 3 is set to 1 MΩ impedance.

**CH<x>:OFFSet**

Sets or queries the offset, in volts, that is subtracted from the specified input channel before it is acquired. The greater the offset, the lower on the display the waveform appears. This is equivalent to setting **Offset** in the Vertical menu.

**Group**
Vertical
Related Commands

CH<x>:POSition

Syntax

CH<x>:OFFSET <NR3>

CH<x>:OFFSET?

Arguments

<NR3> is the desired offset in volts. The range is dependent on the scale and the probe attenuation factor. The offset ranges are shown below.

Table 2–25: Offset Ranges for the TDS 400A, 510A, 540B, 600B, 744A, 754A, & 784A (All Channels) and the TDS 520B, 724A, & 782A (Channel 1 & Channel 2) using a 1x Probe

<table>
<thead>
<tr>
<th>CH&lt;x&gt;:SCALE</th>
<th>OFFSET Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mV/div – 99.5 mV/div</td>
<td>±1 V</td>
</tr>
<tr>
<td>100 mV/div – 995 mV/div</td>
<td>±10 V</td>
</tr>
<tr>
<td>1 V/div – 10 V/div</td>
<td>±100 V</td>
</tr>
</tbody>
</table>

Table 2–26: Offset Ranges for the TDS 520B, 724A, & 782A (Aux 1 & Aux 2) using a 1x Probe

<table>
<thead>
<tr>
<th>CH&lt;x&gt;:SCALE</th>
<th>OFFSET Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 mV/div &amp; 100 mV/div</td>
<td>±0.5 V</td>
</tr>
<tr>
<td>500 mV/div &amp; 1 V/div</td>
<td>±5.0 V</td>
</tr>
<tr>
<td>5 V/div &amp; 10 V/div</td>
<td>±50 V</td>
</tr>
</tbody>
</table>

Examples

CH1:OFFSET 0.5E+00

lowers the channel 1 displayed waveform by 0.5 volts.

CH1:OFFSET?

might return 500.0E−3, indicating that the current channel 1 offset is 0.5 volts.
**CH<x>:POSition**

Sets or queries the vertical position of the specified channel. The position value is applied to the signal before digitization. This is equivalent to setting **Position** in the Vertical menu or adjusting the front-panel **Vertical Position** knob.

**Group**
Vertical

**Related Commands**
CH<x>:OFFSet

**Syntax**

```
CH<x>:POSition <NR3>
```

```
CH<x>:POSition?
```

**Arguments**

<NR3> is the desired position, in divisions from the center graticule. The range is ±5 divisions.

**Examples**

CH2:POSITION 1.3E+00
positions the channel 2 input signal 1.3 divisions above the center of the display.

CH1:POSITION?
might return −1.3E+00, indicating that the current position of channel 1 is at −1.3 divisions.

---

**CH<x>:PROBE? (Query Only)**

Returns the attenuation factor of the probe that is attached to the specified channel.

**Group**
Vertical

**Syntax**

```
CH<x>:PROBE?
```

**Returns**

<NR3>
Examples

CH4:PROBE?
might return 100.0E-3 for a 10x probe.

CH<x>:PROBECal? (Query Only)

*TDS 500B, 600B, & 700A*

Queries the probe cal status for the selected channel.

**Group**
Vertical

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

**Outputs**
FAIL signifies that probe cal has failed for the selected channel.
INIT signifies the probe cal has not yet been run for the selected channel
PASS signifies that probe cal has passed for the selected channel.

**Examples**
CH2:PROBECAL?
might return PASS indicating that probe cal has passed for channel 2.

CH<x>:SCAle

Sets or queries the vertical gain of the specified channel. This is equivalent to setting **Fine Scale** in the Vertical menu or adjusting the front-panel **Vertical SCALE** knob.

**Group**
Vertical

**Related Commands**
CH1:VOLts

**Syntax**
CH<x>:SCAle <NR3>
CH<x>:SCAle?
**Arguments**

<NR3> is the gain, in volts per division. The range is 100 mV per division to 1 mV per division when using a 1x probe.

**Examples**

CH4:SCALE  100E-03
sets the channel 4 gain to 100 mV per division.

CH2:SCALE?
might return 1.00E+0, indicating that the current V per division setting of channel 2 is 1 V per division.

**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

**Related Commands**

CH1:SCAle

**Syntax**

CH<x>:VOLts <NR3>

CH<x>:VOLts?

**Examples**

CH4:VOLTS  100E-03
sets the channel 4 gain to 100 mV per division.

CH2:VOLTS?
might return 1.00E+0, indicating that the current V per division setting of channel 2 is 1 V per division.
**CLEARMenu (No Query Form)**

Clears the current menu from the display. This command is equivalent to pressing the CLEAR MENU button on the front panel.

**Group**
Display

**Syntax**
CLEARMenu

**Examples**
CLEARMenu clears the menu from the display.

**CLS (No Query Form)**

Clears the digitizing oscilloscope status data structures.

**Group**
Status and Error

**Related Commands**

**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 an <E01>, 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 thus MAV. *CLS does not clear the output queue or MAV. (A complete discussion of these registers and bits, and of event handling in general, begins on page 3–1.)
*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a hardcopy output or single sequence acquisition operation is still being processed when the *CLS command is executed.

**CURSor? (Query Only)**

Returns all current cursor settings.

**Group**  
Cursor

**Syntax**  
CURSor?

**Examples**  
CURSOR? might return:

```
:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;
POSITION1 500.0E-6;POSITION2 4.50E-3;SELECT CURSOR1;
:CURSOR:HBARS:POSITION1 3.20E+0;POSITION2 -3.20E+0;
SELECT CURSOR1 as the current cursor settings.
```  

**CURSor:FUNCtion**

Selects and displays the cursor type. Cursors are attached to the selected waveform. This command is equivalent to setting `Function` in the Cursor menu.

**Group**  
Cursor

**Related Commands**  
SELect:CONTROI

**Syntax**  
CURSor:FUNCTION { HBAr s | OFF | VBAr s | PAIred }

CURSor:FUNCTION?
Arguments
HBArs specifies horizontal bar cursors that measure volts.
OFF removes the cursors from the display.
VBAr specifies vertical bar cursors that measure time.
PAIred specifies paired cursors that measure both time and volts.

Examples
CURSOR:FUNCTION VBARS
selects vertical bar type cursors.

CURSor:HBArs? (Query Only)

Returns the current settings for the horizontal bar cursors.

Group Cursor

Syntax CURSor:HBArs?

Examples CURSOR:HBArs?
might return :CURSOR:HBArs:POSITION1 0;POSITION2 0;SELECT CURSOR1.

CURSor:HBArs:DELTa? (Query Only)

Returns the voltage difference between the two horizontal bar cursors.

Group Cursor

Syntax CURSor:HBArs:DELTa?
CURSor:HBArs:POSITION<x>

Positions a horizontal bar cursor.

**Group**  
Cursor

**Syntax**  
CURSor:HBArs:POSITION<x> <NR3>
CURSor:HBArs:POSITION<x>?  

**Arguments**  
<NR3> specifies the cursor position relative to ground, in volts.

**Examples**  
CURSOR:HBArs:POSITION1 25.0E-3  
positions one of the horizontal cursors at 25.0 mV.

CURSOR:HBArs:POSITION2?  
might return -64.0E-3, indicating that one of the horizontal bar cursors is at -64.0 mV.

CURSor:HBArs:POSITION<x>Pcnt  
*TDS 400A Only*

Sets or queries the position of the horizontal bar cursors (x is either 1 or 2) in units of % of vertical range.

**Group**  
Cursor

**Syntax**  
CURSor:HBArs:POSITION<x>Pcnt <NR3>
CURSor:HBArs:POSITION<x>Pcnt?

Arguments

<NR3> has a range of 0 to 100 (%). It specifies the cursor position relative to the vertical range of the selected waveform.

Examples

CURSOR:HBAR:POSITION1PCNT?
might return 4.50 E+1, indicating cursor 1 is positioned at 45% of the vertical range of the selected waveform.

CURSor:HBArs:SELeCt

Selects which horizontal bar cursor is active for front-panel control. The active cursor will be displayed as a solid horizontal line and can be moved using the front-panel general purpose knob when the cursor menu is active. The unselected cursor will be displayed as a dashed horizontal line. This command is equivalent to pressing the SELECT button on the front panel when the Cursor menu is displayed.

Group

Cursor

Syntax

CURSor:HBArs:SELeCt { CURSOR1 | CURSOR2 }
CURSor:HBArs:SELeCt?

Arguments

CURSOR1 selects the first horizontal bar cursor.
CURSOR2 selects the second horizontal bar cursor.
**Examples**

CURSOR:HBARS:SELECT CURSOR1
selects the first horizontal bar cursor as the active cursor.

CURSOR:HBARS:SELECT?
returns CURSOR1 when the first cursor is the active cursor.

**CURSor:HBArs:UNITS**

Sets or queries the units for the horizontal bar cursors. This command is equivalent to setting **Amplitude** in the Cursor menu.

**Group**

Cursor

**Syntax**

CURSOR:HBARS:UNITS { BASE | IRE }

CURSOR:HBARS:UNITS?

**Arguments**

BASE specifies base as the unit of measure.

IRE specifies IRE as the unit of measure. These units are typically used with video signals.

**Examples**

CURSOR:HBARS:UNITS BASE
sets the units for the horizontal bar cursors to base.

CURSOR:HBARS:UNITS?
returns IRE when the horizontal bar cursor units are IRE.

**CURSor:MODe**

Selects whether the two cursors move together in unison or separately.

**Group**

Cursor

**Related Commands**

CURSOR:FUNCTION
**Syntax**

CURSor:MODE { TRACK | INDependent }

CURSor:MODE?

**Arguments**

TRACK ties the two cursors together as you move the general purpose knob.

INDependent frees the two cursors to move separately.

**Examples**

CURSOR:MODE TRACK

specifies that the cursors positions move in unison.

CURSOR:MODE?

might return :TRACK showing the two cursors move in unison.

**CURSor:PAIred**

Positions the paired cursors. Also, returns the current paired cursor settings.

**Group**

Cursor

**Related Commands**

DATa:STARt, DATa:STOP

**Syntax**

CURSor:PAIred SNAP

CURSor:PAIred?

**Arguments**

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

**Examples**

CURSOR:PAIRED SNAP

specifies the positions of the cursors are at the current DATA:START and DATA:STOP values.
**CURSor:PAIred:HDELTA (Query Only)**

Queries the hbar (voltage) distance between the first and second paired cursor. This is the absolute value of the vertical position of the first cursor minus the vertical position of the second cursor.

**Group**  
Cursor

**Related Commands**  
CURSor:FUNCtion

**Syntax**  
CURSor:PAIred:HDELTA?

**Examples**  
CURSor:PAIred:HDELTA?
might return 5.08E+0 for the voltage difference between the two cursors.

**CURSor:PAIred:HPOS1 (Query Only)**

Queries the horizontal bar (voltage) position of the first paired cursor.

**Group**  
Cursor

**Related Commands**  
CURSor:FUNCtion

**Syntax**  
CURSor:PAIred:HPOS1?

**Examples**  
CURSor:PAIred:HPOS1?
might return −64.0E−3, indicating that the first cursor is at −64.0 mV.
CURSor:PAIred:HPOS2 (Query Only)

Queries the horizontal bar (voltage) position of the second paired cursor.

Group Cursor

Related Commands CURSor:FUNCtion

Syntax CURSor:PAIred:HPOS2?

Examples CURSOR:PAIRED:HPOS2?

might return -64.0E-3, indicating the second cursor is at –64.0 mV.

CURSor:PAIred:POSITION<x>

Sets or queries the vertical bar (time) position of the first or second paired cursor.

x is either 1 or 2 and refers to the first or second cursor.

The CURSor:VBArs:UNITS command specifies the units for these cursors.

Group Cursor

Related Commands CURSor:FUNCtion, CURSor:VBArs:UNITS

Syntax CURSor:PAIred:POSITION <NR3>
CURSor:PAIred:POSITION?

<NR3> specifies the cursor position in the units specified by the CURSor:VBArs:UNITS command.

The position can appear in units of seconds, 1/seconds (hertz), or video line numbers (with option 05). On the TDS 400A, position can also appear in units of clocks or 1/clocks.
**Examples**

CURSOR:PAIRED:POSITION1 9.00E-6
specifies the first paired cursor is at 9 μs.

CURSOR:POSITION2?
might return 1.00E-6, indicating that the second paired cursor is at 1 μs.

**CURSor:PAIred:POSITION<x>Pcnt**

*TDS 400A Only*

Sets or queries the horizontal position of the paired cursors (x is either 1 or 2) in units of % of record length.

**Group**  
Cursor

**Syntax**

CURSOR:PAIRED:POSITION<x>Pcnt <NR3>

CURSOR:PAIRED:POSITION<x>Pcnt?

**Arguments**

<NR3> has a range of 0 to 100 (%). It specifies the cursor position relative to the record length of the selected waveform.

**Examples**

CURSOR:PAIRED:POSITION1PCNT?
might return 4.50 E+1, indicating cursor 1 is positioned at 45% of the record length of the selected waveform.

**CURSor:PAIred:SELect**

Selects the active paired cursor. The active cursor appears as a solid vertical line. The unselected cursor appears as a dashed vertical line. This command is equivalent to pressing the SELECT button on the front panel when the Cursor menu is displayed.

**Group**  
Cursor
**Syntax**

CURSor:PAIred:SELect { CURSOR1 | CURSOR2 }

CURSor:PAIred:SELect?

**Arguments**

CURSOR1 specifies the first paired cursor.

CURSOR2 specifies the second paired cursor.

**Examples**

CURSOR:PAIred:SELECT CURSOR2

selects the second paired cursor as the active cursor.

CURSOR:PAIred:SELECT?

returns CURSOR1 when the first paired cursor is the active cursor.

**CURSor:PAIred:UNITS**

Sets or queries the units for the paired cursors. This is equivalent to setting *Amplitude* in the Cursor menu.

**Group**

Cursor

**Syntax**

CURSor:PAIred:UNITS { BASE | IRE }

CURSor:PAIred:UNITS?

**Arguments**

BASE specifies base as the unit of measure.

IRE specifies IRE as the unit of measure. These units are typically used with video signals.
**Examples**

`CURSOR:PAIRED:UNITS BASE`
sets the units for the paired cursors to base.

`CURSOR:PAIRED:UNITS?`
returns IRE when the paired cursor units are IRE.

---

**CURSor:PAIred:VDELTA (Query Only)**

Queries the vbar (time) distance between paired cursors. It returns the absolute value of the first cursor less the second cursor horizontal positions.

The position can appear in units of seconds, 1/seconds (hertz), or video line numbers (with option 05). On the TDS 400A, position can also appear in units of clocks or 1/clocks.

**Group**

Cursor

**Related Commands**

CURSor:FUNCtion

**Syntax**

`CURSOR:PAIred:VDELTA?`

---

**Examples**

`CURSOR:PAIred:VDELTA?`
might return `1.064E+00`, indicating that the time between the paired cursors is 1.064 seconds.

---

**CURSor:VBArts**

Positions the vertical bar cursors and the CURSOR:VBArts? query returns the current vertical bar cursor settings for horizontal position, delta, cursor selection, and units.

The position can appear in units of seconds, 1/seconds (hertz), or video line numbers (with option 05). On the TDS 400A, position can also appear in units of clocks or 1/clocks.

**Group**

Cursor
**DATa:STARt, DATa:STOP, MEASUrement:GAting**

**Syntax**

CURSor:VBArs SNAP

CURSor:VBArs?

**Arguments**

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

**Examples**

CURSOR:VBARS SNAP

specifies that the cursors positions are the same as the current DATA:START and DATA:STOP values.

CURSOR:VBARS?

might return :CURSOR:VBARS:UNIts SECONDS;POSITION1 1.00E-6;POSITION2 9.00E-6;SELECT CURSOR2.

---

**CURSor:VBArs:DELTa? (Query Only)**

Returns the time or frequency between the two vertical bar cursors. The units, seconds or Hertz, are specified by the CURSor:VBArs:UNIts command.

The position can appear in units of seconds, 1/seconds (hertz), or video line numbers (with option 05). On the TDS 400A, position can also appear in units of clocks or 1/clocks.

**Group**

Cursor

**Related Commands**

CURSor:VBArs:UNIts

**Syntax**

CURSor:VBArs:DELTa?

**Returns**

<NR3>
**Examples**

CURSOR:VBARS:DELTa?

might return 1.064E+00, indicating that the time between the vertical bar cursors is 1.064 seconds.

---

**CURSor:VBArs:POSITION<x>**

Positions a vertical bar cursor for both vertical bar and paired cursors. The CURSor:VBArs:UNIts command specifies units.

The position can appear in units of seconds, 1/seconds (hertz), or video line numbers (with option 05). On the TDS 400A, position can also appear in units of clocks or 1/clocks.

**Group**

Cursor

**Related Commands**

CURSor:VBArs:UNIts

**Syntax**

CURSor:VBArs:POSITION<x> <NR3>

CURSor:VBArs:POSITION<x>?

**Arguments**

<NR3> specifies the cursor position in the units specified by the CURSor:VBArs:UNIts command. The position is relative to the trigger position.

**Examples**

CURSOR:VBARS:POSITION2 9.00E–6

defines the vertical bar cursors at 9 μs.

CURSOR:VBARS:POSITION1?

might return 1.00E–6, indicating a vertical bar cursor is at 1 μs.

---

**CURSor:VBArs:POSITION<x>Pcnt**

*TDS 400A Only*

Sets or queries the position of the vertical bar cursors (x is either 1 or 2) in units of % of record length.
**CURSor:VBArs:POSITION**

Selects which vertical bar cursor is active. The active cursor will be displayed as a solid vertical line and can be moved using the front-panel general purpose knob when the cursor menu is active. The unselected cursor will be displayed as a dashed vertical line. This command is equivalent to pressing the **SELECT** button on the front panel when the Cursor menu is displayed.

**Syntax**

```plaintext
CURSor:VBArs:POSITION<x>Pcnt <NR3>
CURSor:VBArs:POSITION<x>Pcnt?
```

**Arguments**

- `<NR3>` has a range of 0 to 100 (%). It specifies the cursor position relative to the record length of the selected waveform.

**Examples**

```plaintext
CURSor:VBArs:POSITION1PCNT?
```

might return 4.50 E+1, indicating cursor 1 is positioned at 45% of the record length of the selected waveform.

**CURSor:VBArs:SELeクト**

Selects which vertical bar cursor is active. The active cursor will be displayed as a solid vertical line and can be moved using the front-panel general purpose knob when the cursor menu is active. The unselected cursor will be displayed as a dashed vertical line. This command is equivalent to pressing the **SELECT** button on the front panel when the Cursor menu is displayed.

**Syntax**

```plaintext
CURSor:VBArs:SELeクト { CURSOR1 | CURSOR2 }
CURSor:VBArs:SELeクト?
```

**Arguments**

- CURSOR1 specifies the first vertical bar cursor.
CURSOR2 specifies the second vertical bar cursor.

**Examples**

```plaintext
CURSOR:VBAR:SELECT CURSOR2
```

selects the second vertical bar cursor as the active cursor.

```plaintext
CURSOR:VBAR:SELECT?
```

returns CURSOR1 when the first vertical bar cursor is the active cursor.

**CURSor:VBArs:UNITS**

Sets or queries the units for the vertical bar cursors. This command is equivalent to setting **Time Units** (**Horiz Units** in the TDS 400A) in the Cursor menu.

**Group**

Cursor

**Related Commands**

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

**Syntax**

```plaintext
CURSor:VBArs:UNITS \{ SECONDS | HErz | LINE (with option 05) | BASE (TDS 400A only) | INVert (TDS 400A only) \}
```

```plaintext
CURSor:VBArs:UNITS?
```

For the TDS 510A, 500B, 600B, & 700A:

For the TDS 400A:
Arguments

SEConds specifies seconds as the unit of measure. For the TDS 400A, this argument is available only for backward compatibility. If a TDS 400A receives this argument, it will convert it to BASE or INVERT depending on the selected waveform. The TDS 400A will not output this argument in response to a query.

HERtz specifies hertz as the unit of measure. For the TDS 400A, this argument is available only for backward compatibility. If a TDS 400A receives this argument, it will convert it to BASE or INVERT depending on the selected waveform. The TDS 400A will not output this argument in response to a query.

LINE specifies a video line as the unit of measure. This is most useful if you have option 05 video trigger installed. On some models, if you do not have option 05, use of this argument will generate an error message. On the TDS 400A, if you do not have option 05, the TDS 400A will use the NTSC standard for converting from time to lines. In order for the measurement to be accurate in the TDS 400A when TRIGGER:MAIN:VIDEO:SYSTEM is set to custom, you must set TRIGGER:MAIN:VIDEO:SCANPERIOD.

BASE (TDS 400A only) specifies the base (or natural) units of the waveform as the unit of measurement. If internal clock is enabled, the base units are seconds. If external clock is enabled, the base units are clocks.

INVERT (TDS 400A only) specifies 1/ base (or natural) units of the waveform as the unit of measurement. If internal clock is enabled, the base units are 1/seconds (hertz). If external clock is enabled, the base units are 1/clocks.

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.
CURSor:VBAr:sUNITStRing? (Query Only)

TDS 400A Only

Queries the unit string as shown on screen for the vertical bar cursor readout.

Group
Cursor

Syntax
CURSor:VBAr:sUNITStRing?

Examples
CURSor:VBAr:sUNITStRing?
might return "ms", indicating the units are milliseconds.

CURVe

Transfers waveform data to and from the digitizing oscilloscope in binary or ASCII format. Each waveform that is transferred has an associated waveform preamble which contains information such as data format and scale. Refer to the WFMPre command starting on page 2–291 for information about the waveform preamble. The data format is specified by the DATa:ENCdg and DATa:WIDTH commands.

The CURVe? query transfers data from the instrument. The data source is specified by the DATa:SOURce command. If more than one source is specified, a comma separated list of data blocks is returned. The first and last data points that are transferred are specified by the DATa:STARt and DATa:STOP commands.

The CURVe command transfers waveform data to the instrument. The data is stored in the reference memory location specified by DATa:DESTination starting with the data point specified by DATa:STARt. Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

A description of the waveform transfer process starts on page 2–32.

Group
Waveform

Related Commands
DATa, WFMPre

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

**Arguments**

<Block> is the waveform data in binary format. The waveform is formatted as: 
\[ \#<x><yyy><data><newline> \] where <x> is the number of y bytes. For example, if <yyy> = 500, then <x> = 3. <yyy> is the number of bytes to transfer. If width is 1 then all bytes on the bus are single data points. If width is 2 then all bytes on the bus are 2-byte pairs. Use the DATa:WIDth command to set the width. <data> is the curve data. <newline> is a single byte newline character at the end of the data. See the GETWFM.C or GETWFM.BAS examples in the accompanying disk for more specifics.

<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, for ASCII data: CURVE

0,0,0,0,-1,1,0,-1,0,0,-1,0,0,-1,0,-1,0,0,-1,1,0,0,-1,0,0,-1,0,0,-1,0,0,-1,0,0,-1,0,0,
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

Related Commands  
CURVE, WAVFrm

Syntax  
DATa { INIT | SNAP }

DATa?

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

SNAP sets DATa:STARt and DATa:STOP to match the current vertical bar cursor positions.

Examples  
DATa SNAP
assigns the current position of the vertical bar cursors to DATa:START and DATa:STOP.

DATa?
might return the string :DATA:ENCOD RPBINARY;DESTINATION REF4; SOURCE REF4;START 1;STOP 500;WIDTH 2

DATa:DESTination

Sets or queries the reference memory location for storing waveform data that is transferred into the digitizing oscilloscope by the CURVe command. This command is identical to the DATa:TARget command.

Group  
Waveform
**DATa:DESTination**

Syntax

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

Arguments

REF<x> (REF1, REF2, REF3 or REF4) is the reference memory location where the waveform will be stored.

Examples

```
DATa:DESTINATION REF3
stores incoming waveform data in reference memory 3.

DATa:DESTINATION?
might return REF2 as the reference memory location that is currently selected.
```

**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 2–27. Setting the DATa:ENCdg value causes the corresponding WFMPre values to be updated and vice versa.

Group

Waveform

Related Commands

WFMPre:ENCdg, WFMPre:BN_FMT, WFMPre:BYT_Or

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 2–27: DATa and WFMPre Parameter Settings

<table>
<thead>
<tr>
<th>DATa:ENCdg Setting</th>
<th>WFMPre Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>:ENCdg</td>
</tr>
<tr>
<td>ASCII</td>
<td>ASC</td>
</tr>
<tr>
<td>RIBinary</td>
<td>BIN</td>
</tr>
<tr>
<td>RPPBinary</td>
<td>BIN</td>
</tr>
<tr>
<td>SRIbinary</td>
<td>BIN</td>
</tr>
<tr>
<td>SRIbinary</td>
<td>BIN</td>
</tr>
</tbody>
</table>

**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.

### DATa:SOURce

Sets or queries the location of the waveform data that is transferred from the instrument by the CURve? query. The source data is always transferred in a predefined order regardless of the order they are specified using this command. The predefined order is CH1 through CH4, MATH1 through MATH3, and then REF1 through REF4.

**Group** Waveform

**Syntax** DATa:SOURce <wfm>[<Comma><wfm>]...

DATa:SOURce?

**Arguments** <wfm> is the location of the waveform data that will be transferred from the digitizing oscilloscope to the controller.
specifies that four waveforms will be transferred in the next CURVE? query. The order that the data will be transferred is CH1, CH2, MATH1, and then REF2.

DATA:SOURCE?
might return REF3, indicating the source for the waveform data that is transferred using a CURVE? query.

**DATa:STARt**

Sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the digitizing oscilloscope.

**Group**
Waveform

**Related Commands**
CURVe?, DATa SNAP, DATa:STOP

**Syntax**

```
DATa:STARt <NR1>
```

```
DATa:STARt?
```

**Arguments**

<NR1> ranges from 1 to the record length, and is the first data point that will be transferred. Data will be transferred from <NR1> to DATa:STOP or the record length, whichever is less. If <NR1> is greater than the record length then no data will be transferred.

When DATa:STOP is less than DATa:STARt, the stop value will equal DATa:STARt + (DATa:STARt – DATa:STOP). For example, if DATa:STARt = 30 and DATa:STOP = 20, then the range of data points for the waveform transfer will equal 30 through 40.

**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.
DATa:STOP

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

When using the CURVe command, the digitizing oscilloscope will stop reading data when there is no more data to read or when the specified record length has been reached so this command will be ignored.

**Group**  
Waveform

**Related Commands**  
CURVe?, DATa SNAp

**Syntax**  
DATa:STOP <NR1>

DATa:STOP?

**Arguments**  
<NR1> ranges from 1 to the record length, and is the last data point that will be transferred. If <NR1> is greater than the record length then data will be transferred up to the record length. If both DATa:STARt and DATa:STOP are greater than the record length, an execution error will be executed.

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

When DATa:STOP is less than DATa:STARt, the stop value will equal DATa:STARt + (DATA:STARt – DATa:STOP). For example, if DATa:STARt = 30 and DATaSTOP = 20, then the range of data points for the waveform transfer will equal 30 through 40.

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

DATA:STOP?  
might return 14900 as the last data point that will be transferred.

DATa:TARget

Sets or queries the location for storing waveform data transferred to the instrument using the CURVe command. This command is equivalent to the
DATa:DESTINATION command and is included for compatibility with older Tektronix instruments.

**Group**  
Waveform

**Related Commands**  
CURVe

**Syntax**  
DATa:TARget REF<x>
DATa:TARget?

**DATa:WIDth**  
Sets the number of bytes per data point in the waveform transferred using the CURVe command.

**Group**  
Waveform

**Related Commands**  
CURVe, WFMPre:BIT_Nr, WFMPre:BYT_Nr

**Syntax**  
DATa:WIDth <NR1>
DATa:WIDth?

**Arguments**  

<NR1> = 1 specifies that there is 1 byte (8 bits) per point. This format is useful when the acquisition mode is set to SAMple, ENVelope, or PEAKdetect (one byte per point). If used for AVERage or HIRes (two bytes per point), the low order byte is not transmitted.

<NR1> = 2 specifies that there are 2 bytes (16 bits) per point. This format is useful for AVERage and HIRes (two bytes per point) waveforms. If used for ENVelope, PEAKdetect, or SAMple (one byte per point), the least significant byte is always zero.
**Examples**  
DATA:WIDth 1  
sets the data width to 1 byte per data point for CURVe data.

**DATE**

Sets or queries the date that the digitizing oscilloscope can display.

**Group**  
Miscellaneous

**Related Commands**  
DISplay: CLOCk, TIMe

**Syntax**  
DATE <QString>  
DATE?

**Arguments**  
<QString> is a date in the form "yyyy-mm-dd".  
mm refers to a two-digit month number from 01 to 12.  
dd refers to a two-digit day number in the month.  
yyyy refers to a four-digit year number.  
There must a dash (–) after the yyyy and after the mm.

**Examples**  
DATE "1993–01–24"  
specifies that the date is set to January 24th, 1993.

**DDT**

Allows the user to specify a command or a list of commands that are executed when the instrument receives a *TRG command or the GET GPIB interface message. *DDT is just a special alias that *TRG uses.

**Group**  
Miscellaneous

**Related Commands**  
ALIAS:DEFINE, *TRG

**Syntax**  
*DDT { <Block> | <QString> }
**DDT?**

Arguments

<Block> or <Qstring> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands (see page 2–4). The sequence must be \( \leq 80 \) characters. <Block> format is always returned as a query response.

Examples

*DDT #0ACQUIRE:STATE RUN;BELL<EOI>

specifies that the acquisition system will be started and the bell rings each time a *TRG command is sent.

**DELEte:SETUp (No Query Form)**

Removes stored setups from memory and initializes the location with the factory default setup.

**NOTE**. The setup information cannot be recovered once it has been deleted.

Group

Save and Recall

Related Commands

*RCL, RECALL:SETUp, *RST, *SAV, SAVe:SETUp, TEKSecure

Syntax

DELEte:SETUp \{ <NR1> | ALL \}

Arguments

<NR1> is a value in the range 1 to 10, and specifies a setup storage location. Using an out-of-range value causes an execution error.

ALL specifies all the stored setups.
Examples

DELETE: SETUP ALL
removes all stored setups. All ten storage locations are initialized to the factory default setup.

DELEte:WA VEform (No Query Form)

Deletes one or all of the stored reference waveforms from memory. The memory allocated for the reference location is then available for reallocation. Memory must be reallocated for the deleted references before any waveform data can be stored in the reference location.

**NOTE.** The waveform data is not actually cleared from the reference location.

Group
Save and Recall

Related Commands
RECAll:WA VEform, SAVe:WA VEform, TEKSecure

Syntax
DELEte:WA VEform { REF<x> | ALL }

Arguments
REF<x> (REF1, REF2, REF3 or REF4) specifies one of the reference memory locations.
ALL specifies all the stored waveforms.

Examples
DELETE:WAVEFORM ALL
removes all the waveforms stored in reference memory.

DELETE:WAVEFORM REF2
removes the waveform stored at REF2.

DESE

Sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see page 3–1.
Group  Status and Error


Syntax  DESE <NR1>

DESE?

Arguments  <NR1> is a value in the range from 0 to 255. The binary bits of the 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, etc.).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER 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 *ESE command to set the 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 the DESER contains the binary value 1011010.

DIAg:RESUIt:FLAg? (Query Only)

Returns the pass/fail status from the last diagnostic test sequence execution. The DIAg:RESUIt:LOG? query can be used to determine which test(s) has failed.

Group  Calibration and Diagnostic
Related Commands

DIAg:RESUlt:LOG?

Syntax

DIAg:RESUlt:FLAg?

Returns

PASS indicating that all of the selected diagnostic tests have passed.
FAIL indicating that at least one of the selected diagnostic tests has failed.

Examples

DIAg:RESULT:FLAg?
returns either PASS or FAIL.

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

Related Commands
DIAg:RESUlt:FLAg?

Syntax

DIAg:RESUlt:LOG?

Returns

QString in the following format:

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

Examples

DIAg:RESULT:LOG?
might return :DIAg:RESULT:LOG "pass--Processor,pass--Display,
pass--FP/Proc Interface,FAIL--Front Panel"
**DIAg:SELect:ACQUISition (No Query Form)**

Selects the acquisition system test sequence that will be run when the DIAg:STATE EXECUte command is sent. This command is equivalent to setting Area in the Utility menu when System is set to Diag/Err.

**Group** Calibration and Diagnostic

**Syntax** DIAg:SELect:ACQUISition ALL

**Arguments** ALL selects functional, memory, and register tests.

---

**DIAg:SELect:ALL (No Query Form)**

Specifies that all system test sequences will be run when the DIAg:STATE EXECUte command is sent. This command is equivalent to setting Area in the Utility menu when System is set to Diag/Err.

**Group** Calibration and Diagnostic

**Syntax** DIAg:SELect:ALL ALL

**Arguments** ALL selects functional, memory, and register tests for the acquisition, processor and display systems, and self diagnostics for the front panel.

---

**DIAg:SELect:CPU (No Query Form)**

Selects the processor system test sequence that will be run when the DIAg:STATE EXECUte command is sent. This command is equivalent to setting Area in the Utility menu when System is set to Diag/Err.

**Group** Calibration and Diagnostic

**Syntax** DIAg:SELect:CPU ALL
Arguments  

**DIAg:SELect:DISplay (No Query Form)**

Selects the display system test sequence that will be run when the DIAg:STATE EXECUte command is sent. This command is equivalent to setting Area in the Utility menu when System is set to Diag/Err.

**Group**  
Calibration and Diagnostic

**Syntax**  
DIAg:SELect:DISplay ALL

Arguments  

**DIAg:SELect:FPAnel (No Query Form)**

Selects the front-panel test sequence that will be run when the DIAg:STATE EXECUte command is sent. This command is equivalent to setting Area in the Utility menu when System is set to Diag/Err.

**Group**  
Calibration and Diagnostic

**Syntax**  
DIAg:SELect:FPAnel ALL

Arguments  

ALL selects functional, memory, and register tests.

ALL selects self diagnostic tests.
DIAg:STATE (No Query Form)

Executes the diagnostic tests that have been specified with the DIAg:SELect commands.

When the test sequence has completed, any of the modules or module interfaces that failed diagnostics are displayed on the screen and stored in an internal log file. The pass/fail status will be returned by the DIAg:RESUlt:FLAg? query, and the internal log will be returned by the DIAg:RESUlt:LOG? query. This command is equivalent to running Extended Diagnostics by selecting **Execute** in the Utility menu when **System** is set to Diag/Err.

**NOTE.** The DIAg:STATE EXECute command can take 30 seconds or more to respond. This command performs a warm boot and does not return control to the instrument controller until diagnostics are complete.

**Group**
Calibration and Diagnostic

**Syntax**
DIAg:STATE EXECute

**Arguments**
EXECute runs the diagnostic test sequences specified by the DIAg:SELect commands. When complete, the digitizing oscilloscope will return to the state it was in just prior to the test. If the PON event was enabled before running the tests, a Service Request will be generated. When the Service Request has been received, the pass/fail status of the tests can be returned by executing the DIAg:RESUlt:FLAg? query.

The DIAg:STATE EXECute command clears the following locations:

- the Event Queue
- the Input Queue
- the Status Registers (SESR and SBR)

To enable a power-on event to generate a Service Request, send the following commands before running diagnostics:

- DESE 128
- *ESE 128
- *SRE 32
- *PSC 0

**Examples**

```
DIAg:STATE EXECUTE
```

executes all the diagnostic tests that have been selected.

**DISplay? (Query Only)**

Returns the current display settings.

**Group**
Display

**Syntax**

```
DISplay?
```

**Examples**

```
DISPLAY?
```

might return:
```
:DISPLAY:FORMAT YT;STYLE VECTORS;FILTER SINX;PERSISTENCE 500.0E-3;GRATICULE FULL;TRIGT 1;INTENSITY:OVERALL 85;
WAVEFORM 70;TEXT 60;CONTRAST 150
```

**DISplay:CLOCk**

Controls the display of the date and time. This is equivalent to setting **Display Date/Time** in the Readout Options side menu. The query form returns an ON (1) or an OFF (0).

**Group**
Display

**Syntax**

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

```
DISplay:CLOCk?
```

- **Examples**

```
OFF
ON
<NR1>
```
Arguments

<OFF> or <NR1> = 0 removes the clock from the display.
<ON> or <NR1> ≠ 0 displays the clock on the display.

Examples

DISPLAY:CLOCK ON
sets the display to show time and date.

DISPLAY:CLOCK?
might return 1 indicating that the display shows time and date.

**DI<sup>1</sup>play:COLO<sup>1</sup>r:CONTR<sup>1</sup>ast**

*TDS 6X4B & 700A Only*

Turns on or off the collision contrast option. The TDS will display overlapping lines in a special collision color when this item is turned on. This is equivalent to selecting **Options** from the main Color menu and toggling **Collision Contrast** in the resulting side menu to **ON** or **OFF**.

Group  
Display

Syntax

```plaintext
DI<sup>1</sup>play:COLO<sup>1</sup>r:CONTR<sup>1</sup>ast { OFF | ON | <NR1> }
DI<sup>1</sup>play:COLO<sup>1</sup>r:CONTR<sup>1</sup>ast?
```

Arguments

<OFF> or <NR1> = 0 turns off collision contrast.
<ON> or <NR1> ≠ 0 turns on collision contrast.

Examples

DISPLAY:COLOR:CONTRAST ON
turns on the contrast option.
**DISPLAY:COLOR:MAP:**<item name>: BYCONTents

*TDS 6X4B & 700A Only*

Determines if the color for a math or reference waveform is set to the color assigned to the waveform contents (the constituent waveform) or to a specific color index. This is equivalent to pressing *Map Math Colors* or *Map Reference Colors* on the Color main menu and *Color Matches Contents* (to select the on state) or *Color* (to select the off state) on the resulting side menu.

**Group**  
Display

**Syntax**  

```
DISPLAY:COLOR:MAP:{ MATH1 | MATH2 | MATH3 | REF1 | REF2 | REF3 | REF4 }:BYCONTents { OFF | ON | <NR1> }
```

```
DISPLAY:COLOR:MAP:{ MATH1 | MATH2 | MATH3 | REF1 | REF2 | REF3 | REF4 }:BYCONTENTS?
```

**Arguments**  

- `<OFF>` or `<NR1> = 0` means set the color by index (NOT by contents).
- `<ON>` or `<NR1> ≠ 0` means set the color by contents (NOT by index).

**Examples**  

```
DISPLAY:COLOR:MAP:MATH1:BYCONTENTS ON
```

sets the display of math waveform # 1 to the color of the waveform used to define math waveform # 1. If two waveforms were used to defined math waveform # 1, then the color of the first mentioned waveform is used.

```
DISPLAY:COLOR:MAP:MATH1:BYCONTENTS?
```

might return `ON`. 
**DISplay:COLOr:MAP:<item name>: TO**

*TDS 6X4B & 700A Only*

Defines the color index to use if setting the color for a math or reference waveform to the color assigned to a specific index. This is similar, but not identical, to pressing **Map Math Colors** or **Map Reference Colors** on the Color main menu, pressing **Color** on the resulting side menu, and entering an index with the general purpose knob or the keypad. The difference is that, when you press **Color** on the front panel, you not only select a color index but also state that you wish to select by index rather than by contents. In contrast, when you use this command you only select the index to use. To state you want to select by index, use the DISplay:COLOr:MAP:...BYCONTents command with the OFF argument.

**Group**  
Display

**Syntax**  
DISplay:COLOr:MAP:{ MATH1 | MATH2 | MATH3 | REF1 | REF2 | REF3 | REF4 }:TO { CH1 | CH2 | CH3 | CH4 | MATH | REF }

DISplay:COLOr:MAP:TO?

**Examples**  
DISPLAY:COLOR:MAP:MATH1:TO CH3  
sets the color of math waveform # 1 to that assigned to channel 3. Note: this assumes that **Color Matches Contents** is set to **OFF**.

DISPLAY:COLOR:MAP:MATH1:TO?  
might return CH3.
**Display:COLOR:PALETTE:PERSistance**  
*TDS 6X4B & 700A Only*

Sets the current persistence palette to one of the preset persistence palettes. This is equivalent to selecting Palette from the main Color menu, Persistence Palettes from the resulting side menu, and Temperature, Spectral, or Gray Scale from the next side menu.

**Group**  
Display

**Syntax**

```
Display:COLOR:PALETTE:PERSistance { TEMPERATURE | SPECTRAL | GRAYSCALE }

DISPLAY:COLOR:PALETTE:PERSistance?
```

**Examples**

```
DISPLAY:COLOR:PALETTE:PERSistance TEMPERATURE
```

sets temperature palette as the current persistence palette.

**Display:COLOR:PALETTE:REGular**  
*TDS 6X4B, & 700A Only*

Sets the current palette to one of the preset palettes. This is equivalent to selecting Palette from the main Color menu and Normal, Bold, Hardcopy, or Monochrome from the resulting side menu.

**Group**  
Display

**Syntax**

```
Display:COLOR:PALETTE:REGular { NORMAL | BOLD | HARDCOPY | MONO }

DISPLAY:COLOR:PALETTE:REGular?
```
**Examples**

DISPLAY:COLOR:PALETTE:REGULAR HARDCOPY

sets the current palette to the hardcopy palette.

**DISplay:COLOR:PALEtte:RESETALL (No Query Form)**

*TDS 6X4B & 700A Only*

Restores all palettes to their factory default settings. This is equivalent to selecting *Restore Colors* from the main Color menu and *Reset All Palettes to Factory* from the resulting side menu.

**Group**

Display

**Syntax**

DISPlay:COLOR:PALEtte:RESETALL

**Examples**

DISPLAY:COLOR:PALETTE:RESETALL

resets the palette to the factory default setting.

**DISplay:COLOR:PALette:<palette name>:RESET (No Query Form)**

*TDS 6X4B & 700A Only*

Restores the given palette to its factory defaults. This is useful if you have edited the colors in a palette. This is equivalent to selecting *Restore Colors* from the main Color menu and *Reset Current Palette to Factory* from the resulting side menu.

**Group**

Display

**Syntax**

DISPlay:COLOR:PALEtte:{ NORMa1 | BOLd | HARDCopy | MONo }:RESET
Examples

DISPLAY:COLOR:PALETTE:HARDCOPY:RESET
resets the hardcopy palette to its initial, factory-default settings.

**DISplay:COLOR:PALETTE:<palette name>:<item name>**

*TDSP 6X4B & 700A Only*

Sets the color of a selected item in a selected palette.

**Group**

Display

**Syntax**

DISplay:COLOR:PALETTE:{ NORMAL | MONo | BOLd | HARDCopy }:
{ BACKGROUND | CH1 | CH2 | CH3 | CH4 | MATH | REF | TEXT | SCROLLTEXT | ZONe | COLLision | GRATicule | SCROLLBAR }
{ <NR1>,<NR1>,<NR1> }?

DISplay:COLOR:PALETTE:{ NORMAL | MONo | BOLd | HARDCopy }:
{ BACKGROUND | CH1 | CH2 | CH3 | CH4 | MATH | REF | TEXT | SCROLLTEXT | ZONe | COLLision | GRATicule | SCROLLBAR }?
Arguments

BACKground specifies the display background color.

CH1 specifies the channel 1 waveform and associated text color.

CH2 specifies the channel 2 waveform and associated text color.

CH3 specifies the channel 3 waveform and associated text color.

CH4 specifies the channel 4 waveform and associated text color.

MATH specifies the default math waveforms and associated text color.

REF specifies the default reference waveforms and associated text color.

TEXT specifies the text color in all menus and all readouts not associated with a waveform.

SCROLLTEXT specifies text in the scroll bar. For example, this covers the scroll bars used in the file system.

ZONE specifies the color of the intensified zone on the waveform when the time base is intensified and the color of the real samples when the display style is **Intensified Samples**.

COLLISION specifies the color of the areas where waveforms overlap each other. You can turn collision marking on or off with the **Items, Waveforms, Options, Collision Contrast** menu item.

GRATICULE specifies the color of the graticule. Use the **Settings, Display, Graticule** option to select the graticule type.
SCROLLBAR specifies the color of the scrollbar. For example, you will find a scrollbar in various file system menus.

<NR1>, <NR1>, <NR1> specifies the desired colors in terms of hue, lightness and saturation values.

Hue is the wavelength of light reflected from the surface. It varies continuously along the color spectrum as produced by a rainbow. Values range from 0 to 359. Sample values are: 0 = blue, 60 = magenta, 120 = red, 180 = yellow, 240 = green, 360 = cyan.

Lightness refers to the amount of light reflected from the surface. It varies from black, to the nominal color, to white. Values range from 0 to 100. A value of 0 results in black. A value of 50 provides the nominal color. A value of 100 results in white.

Saturation is the intensity of color. Completely desaturated color is gray. Completely saturated color of any hue is that color at its most intense. Values range from 0 to 100. A value of 100 provides a pure color. A value of 0 provides gray.

Examples
DISPLAY:COLOR:PALETTE:HARDCOPY:CH1 120,50,100
sets the CH1 color of the hardcopy palette to 120 hue, 50 lightness, and 100 saturation. This provides a pure red.

DISplay:FILTer

Sets or queries the type of interpolation to use for the display. This command is equivalent to setting Filter in the Display menu.

Group
Display

Related Commands
DISplay:STYle

Syntax
DISPLAY:FILTER { LINEar | SINX }
DISPLAY:FILTER?
specified linear interpolation where acquired points are connected with straight lines.

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

**Examples**

DISPLAY:FILTER LINEAR
sets the interpolation filter type to linear.

DISPLAY:FILTER?
returns either LINEAR or SINX, indicating the type of interpolation filter.

**DISplay:FORMat**

Sets or queries the 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 one waveform against another. The source pairs that make up an XY trace are predefined and are listed in Table 2–28. Displaying one source causes its corresponding source to be displayed.
Table 2–28: XY Format Pairs

<table>
<thead>
<tr>
<th>X-Axis Source</th>
<th>Y-Axis Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch 1</td>
<td>Ch 2</td>
</tr>
<tr>
<td>Ch 3 (or AUX 1)</td>
<td>Ch 4 (or AUX 2)</td>
</tr>
<tr>
<td>(All models except TDS 410A)</td>
<td>(All models except TDS 410A)</td>
</tr>
<tr>
<td>Ref 1</td>
<td>Ref 2</td>
</tr>
<tr>
<td>Ref 3</td>
<td>Ref 4</td>
</tr>
</tbody>
</table>

YT sets the display to a voltage 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.

**DISplay:GRA ticule**

Selects the type of graticule that will be displayed. This command is equivalent to setting Graticule in the Display menu.

**Group**
Display

**Syntax**

DISplay:GRAticule { CROSSHair | FRAm | FULL | GRId | NTSc | PAL}

DISplay:GRAticule?

**Arguments**

CROSSHair specifies a frame and cross hairs.

FRAm specifies just a frame.
FULL specifies a frame, a grid, and cross hairs.
GRID specifies a frame and a grid.
NTSc specifies a special NTSC frame.
PAL specifies a special PAL frame.

**Examples**

```
DISPLAY:GRATICULE GRID
```

sets the graticule type to display a frame and a grid.

```
DISPLAY:GRATICULE?
```

returns FULL when all graticule elements (grid, frame, and cross hairs) are selected.

---

**DISplay:INStavu:PERSistence**

*TDS 500B & 700A Only*

Selects the persistence mode to use with InstaVu.

**Group**

Display

**Syntax**

```
DISplay:INStavu:PERSistence { INFPer sist | VARper sist }
DISplay:INStavu:PERSistence?
```

**Arguments**

INFPer sist accumulates data points on the display indefinitely.

VARper sist leaves acquired data points on the display for a period of time specified by DISplay:INStavu:VARper sist.

**Examples**

```
DISPLAY:INSTAVU:PERSISTENCE?
```

might return INFPER SIST indicating the infinite persistence mode is on.
**DISplay:**INStavu:**STYle**  
*TDS 500B & 700A Only*  

Selects how the data is displayed with InstaVu.

**Group**  
Display

**Syntax**  
DISplay:**INStavu:**STYle { DOTs | VECTors }
DISplay:**INStavu:**STYle?

**Arguments**  
DOTs displays individual data points.

VECTors connects adjacent data points. Old points are immediately replaced by new ones.

**Examples**  
DISPLAY:INSTAVU:STYLE?
might return DOTs indicating that the display shows individual waveform data points.

**DISplay:**INStavu:**VARpersist**  
*TDS 500B & 700A Only*  

Sets the length of time that data points are displayed when DISplay:**INSta-

vu:**PERSistance is set to VARpersist. This affects the display only.

**Group**  
Display

**Related Commands**  
DISplay:**INStavu:**PERSistence

**Syntax**  
DISplay:**INStavu:**VARpersist <NR3>
DISplay:**INStavu:**VARpersist?
Arguments  <NR3> specifies the time, in seconds, that the waveform points are displayed on the screen. The range is 250 ms to 10 s.

Examples  DISPLAY:INSTAVU:VARPERSIST 3 specifies that the waveform points are displayed on the screen for 3 seconds before they disappear.

**DISPlay:INTENSITy? (Query Only)**

Returns the current intensity settings for different parts of the display.

**Group**  Display

**Syntax**  DISPlay:INTENSITy?

**Examples**  DISPLAY:INTENSITY? might return :DISPLAY:INTENSITY:WAVEFORM 70;TEXT 60 or :DISPLAY:INTENSITY:OVERALL 85;WAVEFORM 70;TEXT 60;CONTRAST 175

**DISPlay:INTENSITy:CONTRast**

*TDS 4X0A, 510A, 5X0B & 6X0B Only*

Sets the relative intensity of the intensified zone on a waveform. This command is equivalent to setting **Contrast** in the Display Intensity side menu.

The command has no effect on limit test templates or intensified samples. They are displayed at a fixed contrast ratio.

**Group**  Display

**Related Commands**  HORizontal:MODe

**Syntax**  DISPlay:INTENSITy:CONTRast <NR1>
DISplay:INTENSITy:CONTRast?

Arguments  
<NR1> ranges from 100 to 250 percent.

Examples  
DISPLAY:INTENSITy:CONTRAST 140
sets the intensity of the intensified portion of a waveform.

DISplay:INTENSITy:OVERAll
*TDS 4X0A, 510A, 5X0B, & 6X0B Only*

Sets the intensity of the entire display. This command is equivalent to setting Overall in the Display Intensity side menu.

Group  Display

Syntax  
DISPLAY:INTENSITy:OVERAll <NR1>
DISPLAY:INTENSITy:OVERAll?

Arguments  
<NR1> ranges from 20 to 100 percent.

Examples  
DISPLAY:INTENSITy:OVERALL 50
sets the intensity of the display to the middle of the range.

DISPLAY:INTENSITy:OVERALL?
might return 75 as the overall display intensity.
**DISplay:INTENSITy:TEXt**

Sets the intensity of the text and the graticule. This command is equivalent to setting **Text/Grat** in the Display Intensity side menu.

**Group** Display

**Syntax**

DISplay:INTENSITy:TEXt <NR1>

DISplay:INTENSITy:TEXt?

**Arguments**

<NR1> ranges from 20 to 100 percent.

**Examples**

DISPLAY:INTENSITY:TEXT 100

sets the intensity of the text to the brightest level.

**DISplay:INTENSITy:WAVEform**

Sets the intensity of the waveforms. This command is equivalent to setting **Waveform** in the Display Intensity side menu.

**Group** Display

**Syntax**

DISplay:INTENSITy:WAVEform <NR1>

DISplay:INTENSITy:WAVEform?

**Arguments**

<NR1> ranges from 20 to 100 percent.

**Examples**

DISPLAY:INTENSITY:WAVEFORM?

might return 60 as the intensity of the waveform.
**DISplay:MODE**

*TDS 500B & 700A Only*

Selects whether or not to turn on InstaVu.

**Group** Display

**Syntax** DISplay:MODE { INSTavu | NORMal }

DISplay:MODE?

Arguments

INSTavu turns on InstaVu. This mode can help you view infrequent deviations in a signal. It only uses a 500 point record length, no averaging, and no enveloping.

When you turn on InstaVu, the TDS turns off any active zoom, autosave, limit test, waveform math, FastFrame, and XY display. The TDS reactivates these features when you turn off InstaVu.

NORMal turns on the normal (non–InstaVu) mode.

**Examples**

DISPLAY:MODE INSTAVU

.turns on InstaVu.

DISPLAY:MODE?

might return INSTAVU indicating that the InstaVu mode is on.

**DISplay:PERSistence**

Sets the length of time that data points are displayed when DISplay:STYle is set to VARpersist while in normal mode (not InstaVu). This affects the display only and is equivalent to setting *Variable Persistence* in the Display Style side menu.

**Group** Display

**Related Commands** DISplay:STYle
Command Descriptions

**Syntax**

`DISPLAY:PERSISTence <NR3>`

`DISPLAY:PERSISTence?`

**Arguments**

<NR3> specifies the length, in seconds, that the waveform points are displayed on the screen. The range is 250 ms to 10 s.

**Examples**

`DISPLAY: PERSISTENCE 3`

specifies that the waveform points are displayed fading for 3 seconds before they completely disappear.

**DISPLAY:STYLE**

Selects how the data is displayed for normal (not InstaVu) mode. This command is equivalent to setting **Style** in the Display menu.

**Group**

Display

**Related Commands**

`DISPLAY: PERSISTence`

**Syntax**

`DISPLAY:STYLE { DOTs | INFPersist | INTENSIFIED | VARPersist | VECTors }`

`DISPLAY:STYLE?`

**Arguments**

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

INFPersist accumulates data points on the display indefinitely. The display is reset when the style or acquisition is reset.
VARpersist leaves acquired data points on the display for a period of time specified by DISplay:PERSistence.

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

INTENSIFied causes the display to show acquired (non-interpolated) samples with brighter or different colored dots than the rest of the waveform.

Examples

DISPLAY:STYLE INFPERSIST
sets the display to indefinitely accumulate data points on the screen.

DISPLAY:STYLE?
might return DOTS indicating that the display shows individual waveform data points with no history.

**DISplay:TRIGBar**

Controls the display of the trigger level indicator bar/s. The indicator bar shows where the trigger will occur, in voltage.

The digitizing oscilloscope will only display the bar if the trigger source is also displayed. If both a main and a delayed trigger are displayed, then two bars will appear. One will accompany each source. If a logic trigger is selected, then multiple bars may appear. If a runt pulse trigger is selected, then two bars may appear. One will show the upper threshold and one the lower threshold.

**Group** Display

**Syntax**

DISplay:TRIGBar { OFF | SHORT | LONG }

DISplay:TRIGBar?

**Arguments**

OFF removes the trigger indicator bar from the display.

SHORT displays, as the indicator, a short arrow at the right side of the graticule for each displayed trigger signal.
LONG displays, as the indicator, a horizontal line in the center of the graticule for each displayed trigger signal.

**Examples**

DISPLAY:TRIGBAR LONG

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

**DISplay:TRIGT**

Controls the display of the trigger point indicator. This is equivalent to setting the **Display ‘T’ @ Trigger Point** in the Readout Options side menu. The query form returns an ON (1) or an OFF (0).

**Group**

Display

**Syntax**

DISPlay:TRIGT { ON | OFF | <NR1> }

**Arguments**

<OFF> or <NR1> = 0 removes the trigger indicator from the display.

<ON> or <NR1> ≠ 0 displays a trigger indicator on each of the displayed waveforms. The trigger indicator is in reverse video for the selected waveform.

**Examples**

DISPLAY:TRIGT ON

sets the display to show trigger indicators.

DISPLAY:TRIGT?

might return 1 indicating that the display shows trigger indicators.

**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.
Status and Error

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

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 *PSČ is 1. If *PSČ 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.

*ESR? (Query Only)

Returns the contents of the Standard Event Status Register (SESР). *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

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

**Related Commands**  

**Syntax**  
EVENT?

**Examples**  
EVENT?  
might return the response :EVENT 110, showing that there was an error in a command header.
The event code and message in the following format:

\[ \langle \text{Event Code} \rangle \langle \text{Comma} \rangle \langle \text{QString} \rangle [\langle \text{Event Code} \rangle \langle \text{Comma} \rangle \langle \text{QString} \rangle \ldots] \]

\[ \langle \text{QString} \rangle ::= \langle \text{Message} \rangle ; [\langle \text{Command} \rangle ] \]

where \( \langle \text{Command} \rangle \) is the command that caused the error and may be returned when a command error is detected by the digitizing oscilloscope. As much of the command will be returned as possible without exceeding the 60 character limit of the \( \langle \text{Message} \rangle \) and \( \langle \text{Command} \rangle \) strings combined. The command string is right-justified.

**Examples**

EOVMSG?

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

**EVQTY? (Query Only)**

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

**Group**

Status and Error

**Related Commands**

ALLEv?, EVENT?, EVMsg?

**Syntax**

EVOty?

**Returns**

\(<\text{NR1}\>)

**Examples**

EVOQTY?

might return 3 as the number of event codes in the Event Queue.
**FACTory (No Query Form)**

Resets the digitizing oscilloscope to its factory default settings. This command is equivalent to selecting **Recall Factory Setup** in the Waveform Save/Recall menu.

**Group**  
Miscellaneous

**Related Commands**  

**Syntax**  
FACTory

Setting the digitizing oscilloscope to factory default includes:

- Clears the Event Status Enable Register.
- Clears the Service Request Enable Register.
- Sets the Device Event Status Enable Register to 255.
- Sets the Power On Status Clear Flag to TRUE
- Purges all defined aliases.
- Enables all Command Headers (HEADer ON).
- Sets the macro defined by *DDT to a “zero-length field.”
- Clears the pending operation flag and associated operations.

The FACTory command does not alter the following:

- The state of the GPIB (IEEE Std 488.1-1987) interface.
- The selected GPIB address.
- Calibration data that affects device specifications.
- Protected user data.
- Stored settings.
- The current password (if implemented).
**FILESystem:COPY (No query form)**

*File System Only*

Copies a named file or files to a new file. The new file may be in a totally separate directory than the old file. Also, you can use wild card characters (*.*) to copy multiple files with one command.

**Group**

File system

**Syntax**

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

**Arguments**

- `<file path>` is a quoted string that defines the file name and path. Input the file path using the form `<drive>/<dir>/<filename>`. `<drive>` and one or more `<dir>`s are optional. If you do not specify them, the TDS will copy the file in the current directory. `<filename>` stands for a filename of up to 8 characters and can be followed by a period (".") and a 3-character extension. You can also use the inclusive filename *.* in the source file path to copy all files.

- `<directory path>` is a quoted string that defines the directory. Input the directory using the form `<drive>/<dir>/<directory name>`. `<drive>` and one or more `<dir>`s are optional. If you do not specify them, the TDS will copy the directory in the current directory. `<directory name>` stands for a directory name of up to 8 characters and can be followed by a period (".") and a 3-character extension.

**Examples**

```
FILESYSTEM:COPY "TEK00001.SET","fd0:/TEK00001.SET"
copies the file named TEK00001.SET on the current drive to a file named TEK00001.SET on the drive fd0: in the root directory.

FILESYSTEM:COPY "fd0:/YOURDIR/TEK00001.SET","fd0:/MYDIR"
copies the file named TEK00001.SET on the drive and the YOURDIR directory to the MYDIR directory on the same drive.
```
FILESYSTEM: COPY "YOURDIR", "fd0:/MYDIR"
copies the files in the YOURDIR directory in the current directory to the MYDIR
directory on the fd0: drive.

**FILESystem:CWD**
*File System Only*

Sets or returns the current working directory (CWD) path.

**Group** File system

**Syntax**

FILESystem:CWD <directory path>

FILESystem:CWD?

**Arguments**

<directory path> is a quoted string that defines the directory name and path.

**Examples**

FILESYSTEM:CWD "fd0:/MYDIR"
will define fd0:/MYDIR as the current directory.

FILESYSTEM:CWD?
might return fd0:/MYDIR if that is the current directory.

**FILESystem:DELEte (No query form)**
*File System Only*

Deletes a named file.

**Group** File system

**Syntax**

FILESystem:DELEte <file path>
Arguments

<file_path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will delete the file in the current directory. <filename> stands for a filename of up to 8 characters and can be followed by a period (".") and a 3-character extension. You can also use the inclusive filename "*.*" to delete all files.

Examples

FILESYSTEM:DELETE "NOT-MINE.SET"
deletes the file named NOT-MINE.SET on the default drive and directory.

FILESYSTEM:DELETE "*.*"
deletes all the files in the default directory on the default drive.

FILESystem:DELWarn

File System Only

Turns on or off the front-panel file-delete warning. No warning is returned via the GPIB.

Group

File system

Syntax

FILESystem:DELWarn { ON | OFF | <NR1> }

FILESystem:DELWarn?

Arguments

ON or <NR1> ≠ 0 turns on the front-panel delete warning.

OFF or <NR1> = 0 turns off the front-panel delete warning.

Examples

FILESYSTEM:DELWARN OFF
disables the front-panel delete warning.

FILESYSTEM:DELWARN?
might return 0 indicating the front-panel warning is disabled.
**FILESystem:DIR (Query only)**

*File System Only*

Returns a list of quoted strings. Each string contains the name of a file or directory in the current directory.

**Group**  
File system

**Syntax**  
FILESystem:DIR?

**Examples**  
FILESYSTEM:DIR?  
returns a list of files and directories in the default directory.

**FILESystem:FORMAT (No query form)**

*File System Only*

 Formats a named drive.

**Group**  
File system

**Syntax**  
FILESystem:FORMAT <drive name>

**Arguments**  
<drive name> is a quoted string that defines the disk drive to format. fd0: refers to the floppy-disk drive built into the digitizing oscilloscope.

**Examples**  
FILESYSTEM:FORMAT "fd0:"  
formats the media on drive fd0:.

**FILESystem:FREESpace (Query only)**

*File System Only*

Returns the amount of freespace (in bytes) on the current drive.
**FILESyste:m:FREESpace?**

**Syntax**

`FILESyste:m:FREESpace?`

**Examples**

FILESyste:m:FREESpace?

might return 0 as the amount of freespace available if the drive was full.

---

**FILESyste:m:MKDir (No query form)**

**File System Only**

Make a new directory.

**Syntax**

`FILESyste:m:MKDir <directory path>`

**Arguments**

`<directory path>` is a quoted string that defines the directory. Input the directory using the form `<drive>/<dir>/<directory name>`. `<drive>` and one or more `<dir>`s are optional. If you do not specify them, the TDS will create the directory in the current directory. `<directory name>` stands for a directory name of up to 8 characters and can be followed by a period (`.`) and a 3-char extension.

**Examples**

FILESyste:m:MKDIR "NEATPICS"

creates the directory named NEATPICS on the current drive.

---

**FILESyste:m:OVERWrite**

**File System Only**

Turns on or off the file-overwrite protection. Turning on file-overwrite protection prevents writing over existing files.
Syntax  
FILESystem:OVERWrite { ON | OFF | <NR1> }

FILESystem:OVERWrite?

Arguments  
ON or <NR1>  ≠  0 turns on the file-overwrite protection.

OFF or <NR1>  =  0 turns off the file-overwrite protection.

Examples  
FILESYSTEM:OVERWRITE OFF
lets you overwrite existing files.

FILESYSTEM:OVERWRITE?
might return 0 indicating you cannot overwrite existing files.

FILESystem:PRInt (No query form)

File System Only

Prints a named file to the named port.

Group  
File system

Syntax  
FILESystem:PRInt <filepath>,{ GPIb | RS232 | CENtronics }

Arguments  
<filepath> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will print the file in the current directory. <filename> stands for a filename of up to 8 characters and can be followed by a period ("."), and a 3-character extension.

GPIb specifies that the hardcopy is sent out the GPIB port.
CENTronics specifies that the hardcopy is sent out the Centronics port (Option 13 RS232/Centronics Hardcopy Interface only).

RS232 specifies that the hardcopy is sent out the RS232 port (Option 13 RS232/Centronics Hardcopy Interface only).

**Examples**

FILESYSTEM:PRINT "TEK00000.IBM",CENTRONICS sends the file named TEK00000.IBM out the Centronics port.

### FILESystem:REName (No query form)

*File System Only*

Assigns a new name to a file.

**Group**

File system

**Syntax**

FILESystem:REName <old file path>,<new file path>

**Arguments**

<old file path> is a quoted string that defines the file to rename. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will look for the filename in the current directory.

<new file path> is a quoted string that defines the new name of the file. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will place the newly named file in the current directory.

**Examples**

FILESYSTEM:RENAME "TEK00000.SET","MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains on the current directory.

### FILESystem:RMDir (No query form)

*File System Only*

Deletes a named directory.

**Group**

File system
Syntax  
FILESystem:RMDir <directory path>

Arguments  
<directory path> is a quoted string that defines the directory. Input the directory using the form <drive>/<dir>/<directory name>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will delete the directory in the current directory. <directory name> stands for a directory name of up to 8 characters and can be followed by a period (".") and a 3-character extension.

Examples  
FILESYSTEM:RMDIR "NEATPICS"  
deletes the directory named NEATPICS in the current directory.

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.

The HARDCopy? query returns format, layout, and port information.

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

Group  
Hardcopy

Syntax  
HARDCopy { ABOrt | CLEARSpool | STARt }  
HARDCopy?

Arguments  
ABOrt terminates the hardcopy output in process.
NOTE. DCL does NOT clear the output queue once a hardcopy is in process. The only way to abort the hardcopy process is to send the HARDCopy ABOrt command. The output queue can then be cleared using DCL.

CLEARspool clears the printer output spooler.

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

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

Examples

HARDCopy ABORT
stops any hardcopy output that is in process.

HARDCopy:FILENAME
File System Only

Selects the file to send the hardcopy data to on the next hardcopy command (HARDCopy START). This is equivalent to setting the target file name in the Hardcopy menu.

Group
Hardcopy

Related Commands
HARDCopy

Syntax

HARDCopy:FILENAME <file path>

HARDCopy:FILENAME?

<file path> specifies that the hardcopy is sent to the named file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>,<drive> and one or more <dir>s are optional. If you do not specify them, the TDS will write the file to the current
directory. `<filename>` stands for a filename of up to 8 characters followed by a period (".") and any 3-character extension.

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

**Examples**

`HARDCOPY:FILENAME "TEK.IBM"`

selects TEK.IBM as the selected file name.

`HARDCOPY:FILENAME?`

might return TEK.IBM as the selected file name.

`HARDCOPY:FILENAME "TEK??.IBM"`

selects TEK as the selected file name with a numeric, two-digit suffix. The TDS might return TEK00.IBM as the first file, TEK01.IBM as the second.

**HARDCopy:FORMat**

Selects the output data format for hardcopies. This is equivalent to setting `Format` in the Hardcopy menu.

**Group**

Hardcopy

**Syntax**

```
HARDCopy:FORMat { BMP | BMPCOLOR | DESKJET | DESKJETC (TDS 500B, 600B, & 700A) | DPU411 | DPU412 | EPSCOLOR | EPSImage | EPSMono | EPSOn | HPG1 | INTERLeaf | LASERJet | PCX | PCXCOLOR | RLE | THInkjet | TIFF }

HARDCopy:FORMat?
```
On monochrome instruments, the following formats are mapped to a monochrome near equivalent:

- PCXCOLOR → PCX
- BMP COLOR → BMP
- RLE → BMP
- EPS COLOR → EPSIMAGE
- DESKJETC → DESKJET

(The DESKJETC argument is not on the TDS 400A and 510A)

For example: if HARDCOPY:FORMAT PCXCOLOR and HARDCOPY:FORMAT? are sent to the oscilloscope, PCX is returned.

**Examples**

HARDCOPY:FORMAT HPGL
sets the hardcopy output format to HPGL.

HARDCOPY:FORMAT?
might return INTERLEAF as the final hardcopy output format.
Command Descriptions

HARDCopy:LAYout

Selects the printing orientation. This is equivalent to setting Layout in the Harccopy menu.

Group Hardcopy

Syntax HARDCopy:LAYout { LANDscape | PORTRait }

HARDCopy:LAYout?

Arguments LANDscape specifies that the bottom of the hardcopy is along the long side of the page.

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

Examples HARDCOPY:LAYOUT?

might return PORTRAIT as the hardcopy page-layout format.

HARDCopy:PALEtte

TDS 644B, 684B, & 700A Only

Selects whether to create the hardcopy using the current color palette (as set in the Display menu and seen on the screen) or the hardcopy palette. For color hardcopies, the default hardcopy palette may be most appropriate, since it has a white background and requires less ink for printing onto white paper. For monochrome hardcopies, the TDS ignores the palette and prints black (or the default color) objects on a blank background.

Group Hardcopy

Related Commands HARDCopy

Syntax HARDCopy:PALEtte { CURRent | HARDCopy }
HARDCopy:PALette

Examples
HARDCOPY:PALETTE HARDCOPY
would print each copy made using the hardcopy palette.

HARDCopy:PORT

Selects where to send the hardcopy data on the next hardcopy command (i.e. HARDCOPY START command). This is equivalent to setting Port in the Hardcopy menu.

Group
Hardcopy

Related Commands
HARDCopy, LIMit:HARDCopy

Syntax
HARDCopy:PORT { CENtronics | FILE (File System only) | GPIb | RS232 }
HARDCopy:PORT?

CENtronics specifies that the hardcopy is sent out the Centronics port (available with the RS232/Centronics Hardcopy Interface).

FILE specifies that the hardcopy is stored in the file named in the HARDCOPY:FILENAME command.

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

RS232 specifies that the hardcopy is sent out the RS232 port (Option 13 RS232/Centronics Hardcopy Interface only).
Examples
HARDCOPY:PORT?
might return GPIB as the selected hardcopy output port.

HDR
This command is identical to the HEADer query and is included for compatibility with older Tektronix instruments.

Group
Miscellaneous

Syntax
HDR { <NR1> | OFF | ON }

HEADer
Sets and queries the Response Header Enable State that causes the digitizing 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

Related Commands
VERBose

Syntax
HEADer { <NR1> | OFF | ON }

HEADer?
0N or \(<\text{NR1}\neq 0\) sets the Response Header Enable State to true. This causes the digitizing oscilloscope to include headers on applicable query responses. You can then use the query response as a command.

0FF or \(<\text{NR1}\leq 0\) sets the Response Header Enable State to false. This causes the digitizing oscilloscope to omit headers on query responses, so that only the argument is returned.

**Examples**

HEADER OFF
causes the digitizing oscilloscope to omit headers from query responses.

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

### HORizontal? (Query Only)

Returns all settings for the horizontal commands. The commands \(\text{HORizontal:MAIN:SCALE}\), \(\text{HORizontal:MAIN:SECdiv}\), \(\text{HORizontal:SCALE}\), and \(\text{HORizontal:SECdiv}\) are equivalent so \(\text{HORizontal:MAIN:SCALE}\) is the only value that is returned.

**Group**
Horizontal

**Syntax**
HORizontal?

**Examples**
HORIZONTAL?
might return the string:

\[ :\text{HORIZONTAL:MODE}\ \text{MAIN};\text{RECORDLENGTH}\ 500; \text{POSITION}\ 5.0E+0;\text{TRIGGER:POSITION}\ 50; :\text{HORIZONTAL:MAIN:SCALE}\ 1.0E-6; :\text{HORIZONTAL:DELAY:MODE}\ \text{RUNSAFTER};\text{SCALE}\ 1.0E-6; \text{TIME}: 16.0E-9 \]
HORizontal:CLK

*TDS 400A Only*

Enables either the internal or external clocks. The query returns whether the clock is set to internal or external.

- **Group**: Horizontal
- **Syntax**: `HORizontal:CLK { INTERNAL | EXTERNAL }

HORizontal:CLK?

- **Arguments**
  - INTERNAL enables the internal clock mode.
  - EXTERNAL enables the external clock mode.

- **Examples**
  - `HORIZONTAL: CLOCK INTERNAL`
  - Enables the internal clocks.

HORizontal:CLK:MAXRate

*TDS 400A Only*

Sets the maximum external clock rate. It does not enable the external clock. The maximum external clock rate affects the decimation rate in Hi-Res mode.

If set to less than or equal to 1000, this command enables roll mode when external clock is on and roll mode is set to auto.

- **Group**: Horizontal
- **Syntax**: `HORizontal:CLK:MAXRate <NR3>

HORizontal:CLK:MAXRate?`
Arguments  

<NR3> is rounded up to the nearest allowable external clock rate.

Examples  

HORIZONTAl:CLOCK:MAXRATE 50

sets the maximum external clock rate to 50 clocks per second.

HORizontal:DELay? (Query Only)

Returns all horizontal delayed time base parameters. The commands HORizontal:DELa:y:SECdiv and HORizontal:DELa:y:SCAle are identical so only HORizontal:DELa:y:SCAle will be returned.

Group  

Horizontal

Related Commands  


Syntax  

HORizontal:DELa:y?

Examples  

HORIZONTAl:DELa:y?

might return the delay parameters :HORIZONTAl:DELa:y:MODE RUNSAF-Ter;SCAle 1.0E-6;TIME: 16.0E-9

HORizontal:DELa:y:MODE

Selects the mode for the delayed time base. This is equivalent to setting Time Base in the Horizontal menu.

Group  

Horizontal

Related Commands  

HORizontal:DELa:y:TIME

Syntax  

HORizontal:DELa:y:MODE { RUNSAfter | TRIGAfter }
HORizontal:DELay:MODE?

Arguments
- RUNSAfter specifies that the delayed time base runs a user-specified amount of delay time after the main trigger event.
- TRIGAfter specifies that the delayed time base is triggerable after the main time base triggers.

Examples
HORizontal:DELay:MODE?
returns either RUNSAFTER or TRIGAFTER, indicating the delayed time base mode.

HORizontal:DELay:SCALE

Sets the time per division for the delayed time base. This is equivalent to setting Delayed Scale in the Horizontal Scale side menu.

On the TDS 400A, changes made while the external clock is enabled do not take effect until the internal clock is enabled. Also, when the external clock is enabled, the query form of this command returns an <NR3> value representing ‘50’.

Group
Horizontal

Related Commands
HORizontal:DELay:SECdiv

Syntax
HORizontal:DELay:SCAle <NR3>
HORizontal:DELay:SCAle?

Arguments
- <NR3> is the time per division. The range is 10 s (5 s on the TDS 620A, 640A, and 644A and 20 s on the TDS 400A) to 500 ps (1 ns on the TDS 400A and 200 ps on the TDS 784A) in a 1–2–5 sequence. Values that are not in a 1–2–5
sequence (1–2.5–5 on the TDS 620A, 640A, & 644A) will be set to the closest valid value. If the delayed time base scale is set slower than the main time base scale, both the main and delayed time base scales will be set to the delay scale value.

**Examples**

```
HORizontal:DELAY:SCALE 2.0E-6
```
sets the delay scale to 2 µs per division.

```
HORizontal:DELAY:SCALE 9.0E-6
```
sets the delay scale to 10 µs per division. Since 9 µs is not a valid value within the 1–2–5 sequence (1–2.5–5 on the TDS 620A, 640A, & 644A), it is automatically set to the closest valid value.

```
HORizontal:DELAY:SCALE?
```
might return 1.0E-3, indicating that the delay time is 1 ms per division.

### 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 digitizing oscilloscopes.

**Group**

Horizontal

**Syntax**

```
HORizontal:DELay:SECdiv <NR3>
HORizontal:DELay:SECdiv?
```

### HORizontal:DELay:TIMe

Sets or queries the delay time to wait after the main trigger before the delayed time base begins. This is equivalent to setting **Delayed Runs After Main** in the **Time Base** side menu of the Horizontal menu.

**Group**

Horizontal

**Related Commands**

HORizontal:DELay:MODe
**HORizontal:DELAY:TIME**  
*<NR3>*

**Syntax**  
HORizontal:DELAY:TIME <NR3>

**Arguments**  
<NR3> is the time, in seconds, between the main trigger and the delayed trigger. The range on the TDS 510A, 500B, 600B, & 700A is from 16 ns to 250 seconds with a resolution of 4 ns. The range on the TDS 400A is from 10 ns to 20 seconds with a resolution of 10 ns.

**Examples**  
HORIZONTAL:DELAY:TIME 2.0E-3
sets the delay time between the main and delayed time base to 2 ms.

**HORizontal:DELAY:TIME? (Query Only)**

Returns the delay time parameters.

**Group**  
Horizontal

**Related Commands**  

**Syntax**  
HORizontal:DELAY:TIME?

**Examples**  
HORIZONTAL:DELAY:TIME?

**HORizontal:DELAY:TIME:RUNSAfter**

Sets or queries the delay time to wait after the main trigger before the delayed time base begins. This is equivalent to setting **Delayed Runs After Main** in the **Time Base** side menu of the Horizontal menu.

**Group**  
Horizontal

**Related Commands**  
HORizontal:DELAY:MODE

**Syntax**  
HORizontal:DELAY:TIME:RUNSAfter <NR3>
**HORizontal:DElay:TIME:RUNSAfter?**

**Arguments**

<NR3> is the time, in seconds, between the main trigger and the delayed trigger. The range is from 16 ns (10 ns on the TDS 400A) to 250 seconds (20 s on the TDS 400A) with a resolution of 4 ns.

**Examples**

HORIZONTAL:DELAY:TIME:RUNSAFTER 2.0E-3
sets the delay time between the main and delayed time base to 2 ms.

**HORizontal:DElay:TIME:TRIGAAfter**

Sets the delay time to wait in the trigger after delay mode. This is the time that must pass before a delayed trigger is accepted. This command is equivalent to setting Delay by Time time in the Delayed Trigger menu.

**Group**

Horizontal

**Related Commands**

HORizontal:DElay:MODe

**Syntax**

HORIZONTAL:DELAY:TIME:TRIGAFTER <NR3>

HORIZONTAL:DELAY:TIME:TRIGAFTER?

**Arguments**

<NR3> is the delay time, in seconds. The range is from 16 ns (60 ns on the TDS 400A) to 250 seconds (20 s on the TDS 400A) with a resolution of 4 ns (10 ns down to 110 ns on the TDS 400A).

**Examples**

HORIZONTAL:DELAY:TIME:TRIGAFTER 4.0E-6
sets the delay time to 4 μs.
Horizontal:FastFrame:count

*TDS 500B & 700A Only*

Sets or queries FastFrame frame count. This is equivalent to setting FastFrame Setup in the Horizontal menu and the Frame Count menu item in the side menu. FastFrame, also known as memory segmentation, captures a series of triggered acquisitions with minimal, intervening, time.

**Group** Horizontal

**Syntax**

HORizontal:FASTframe:COUNt <NR1>

HORizontal:FASTframe:COUNt?

**Arguments**

<NR1> indicates the number of frames to acquire.

**Examples**

HORIZONTAL:FASTFRAME:COUNt 2

Sets up FastFrame mode to acquire two frames (segments) of data.

Horizontal:FastFrame:length

*TDS 500B & 700A Only*

Setup length of each FastFrame frame. This is equivalent to setting FastFrame Setup in the Horizontal menu and the Frame Length menu item in the side menu. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal, intervening, time between them.

**Group** Horizontal

**Syntax**

HORizontal:FASTframe:LENgth <NR1>

HORizontal:FASTframe:LENgth?
Arguments  

<NR1> indicates the frame (segment) length.

Examples  

HORIZONTAL:FASTFRAME:LENGTH 250  
Sets up the FastFrame mode to acquire frames of 250 samples each.

HORizontal:FASTframe:POSition  
*TDS 500B & 700A Only*

Display the selected FastFrame frame. This is equivalent to selecting Horiz Pos in the Horizontal menu, selecting the Frame menu item in the side menu, and entering a value with the keypad or the general purpose knob. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal, intervening, time between them.

Group  

Horizontal

Syntax  

HORizontal:FASTframe:POSition <NR1>  
HORizontal:FASTframe:POSition?

Arguments  

<NR1> indicates the selected frame to display.

Examples  

HORIZONTAL:FASTFRAME:POSITION 25  
Selects the 25th FastFrame frame to display.

HORizontal:FASTframe:STATE  
*TDS 500B & 700A Only*

Setup FastFrame acquisition. This is equivalent to setting FastFrame Setup in the Horizontal menu and the FastFrame menu item in the side menu. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal time between them.
The digitizing oscilloscope in FastFrame mode is ready to accept a continuous burst of triggers 400 ms after the controller sends the acquire:state run command.

**Group**  
Horizontal

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

HORizontal:FASTframe:STATE?

**Arguments**  
<NR1> indicates OFF if it’s a 0 or ON if it’s a 1 (or any other nonzero value).

ON means turn on FastFrame.

OFF means turn off FastFrame.

The query form only returns 0 or 1.

**Examples**  
HORIZONTAL:FASTFRAME:STATE ON

turns on FastFrame.

**HORizontal:FITtoscreen**

Setup horizontal waveform compress operation. This command is equivalent to setting Record Length in the Horizontal menu and the Fit to screen menu item in the side menu. Waveform compress lets you fit a captured waveform to the visible screen. It provides the same functionality as if you were in zoom mode and changed the time per division until the waveform fit the screen.

**Group**  
Horizontal

**Syntax**  
HORizontal:FITtoscreen { <NR1> | OFF | ON }

HORizontal:FITtoscreen?
Arguments  

<NR1> indicates OFF if it’s a 0. It indicates ON if it’s a non-zero value.

ON means turn on waveform compress.

OFF means turn off waveform compress.

Examples  

HORIZONTAIAL:FITTOSCREEN ON

turns on waveform compress.

HORizontal:MAIn? (Query Only)

Returns the time per division of the main time base. The commands HORizontal:MAIn:SECdiv and HORizontal:MAIn:SCAle are identical so only HORizontal:MAIn:SCAle will be returned.

Group  

Horizontal

Related Commands  


Syntax  

HORizontal:MAIn?

Examples  

HORIZONTAIAL:MAIN?

might return :HORIZONTAIAL:MAIN:SCALE 1.0E-6.

HORizontal:MAIn:SCAle

Sets the time per division for the main time base. This command is equivalent to setting Main Scale in the Horizontal Scale side menu.

On the TDS 400A, changes made while the external clock is enabled do not take effect until the internal clock is enabled. Also, when the external clock is enabled, the query form of this command returns an <NR3> value representing ‘50’.
Command Descriptions

**Group**
Horizontal

**Related Commands**

**Syntax**
HORizontal:MAIn:SCAle <NR3>
HORizontal:MAIn:SCAle?

**Arguments**
<NR3> is the time per division. For the TDS 400A series, the range is 20 s to 1 ns. For the TDS 510A, 500B, 600B, and 700A, the range is 10 s to 500 ps (or 200 ps on the TDS 784A), in a 1–2–5 sequence.

**Examples**
HORIZONTAL:MAIN:SCALE 2E-6
sets the main scale to 2 µs per division.

**HORizontal:MAIn:SECdiv**
Sets the time per division for the 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 digitizing oscilloscopes.

**Group**
Horizontal

**Related Commands**

**Syntax**
HORizontal:MAIn:SECdiv <NR3>
HORizontal:MAIn:SECdiv?

**Arguments**
<NR3> is the time per division. For the TDS 400A series, the range is 20 s to 1 ns. For the TDS 510A, 500B, 600B, and 700A, the range is 10 s to 500 ps (or 200 ps on the TDS 784A), in a 1–2–5 sequence.

**Examples**
HORIZONTAL:MAIN:SCALE 2E-6
sets the main scale to 2 µs per division.
HORizontal:MODE

Selects whether the horizontal display uses the main or delayed time base or both. This command is equivalent to setting Time Base in the Horizontal menu.

**Group**
Horizontal

**Related Commands**
DISplay:INTENSITY:CONTRast

**Syntax**
HORizontal:MODE { DELAYed | INTENSIFIED | MAIN }  
HORizontal:MODE?

**Arguments**
DELAYed means that the selected waveform is horizontally scaled relative to the delayed time base.

INTENSIFIED uses both the main and delay scales to display the waveform. The portion of the waveform that would be displayed in DELAYed mode is intensified. The level of intensity is set by the DISplay:INTENSITY:CONTRast command.

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

**Examples**
HORIZONTAL:MODE DELAYED
uses the delayed horizontal scale to display the waveform.

HORIZONTAL:MODE?
might return INTENSIFIED, indicating that the waveform is displayed using both the main and delayed time base scale.
HORizontal:POSition

Positions the waveform horizontally on the display. This is used for both main and delayed time bases. This command is equivalent to adjusting the front-panel Horizontal Position knob or setting the position in the Horizontal Position side menu.

Group    Horizontal

Syntax    HORizontal:POSition <NR3>

HORizontal:POSition?

Arguments  <NR3> is from 0 to 100, and is the percent of the waveform that is displayed left of the center graticule.

Examples  HORIZONTAL:POSITION 10

sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

HORizontal:RECOordlength

Sets the number of data points that are acquired for each record. This is equivalent to setting Record Length in the Horizontal menu.

Group    Horizontal

Syntax    HORizontal:RECOordlength <NR1>

HORizontal:RECOordlength?

Arguments  Table 2–29 shows supported values for <NR1>.
### Table 2–29: Record Length Values (<NR1>)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>500, 1000, 2500, 5000, 15000, 30000</td>
<td>500, 1000, 2500, 5000, 15000, 50000</td>
<td>500, 1000, 2500, 5000, 15000</td>
<td>500, 1000, 2500, 5000, 15000, 50000</td>
</tr>
<tr>
<td>Option 1M (not on the TDS 510A)</td>
<td>60000, 120000</td>
<td>TDS 520B, 724A, &amp; 782A: 75000, 100000, 130000, 250000</td>
<td>75000, 100000, 130000, 250000, 500000</td>
<td></td>
</tr>
</tbody>
</table>

#### Examples

HORIZONTAL:RECORFLN 2500

specifies that 2500 data points will be acquired for each record.

HORIZONTAL:RECORFLN?

might return 15000 as the number of data points per record.

### HORizontal:ROLL

* **TDS 400A Only**

Sets roll mode to auto or off.

**Group**  
Horizontal

**Syntax**  
HORizontal:ROLL { ON | OFF | <NR1> }

HORizontal:ROLL?
Arguments

OFF or <NR1> = 0 turns off roll mode.

ON or <NR1> ≠ 0 turns on roll mode – if the time/division is set appropriately.

**HORizontal:SCAle**

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

**Group**
Horizontal

**Syntax**
HORizontal:SCAle <NR3>
HORizontal: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 here for compatibility purposes.

**Group**
Horizontal

**Syntax**
HORizontal:SECdiv <NR3>
HORizontal:SECdiv?

**HORizontal:TRIGger? (Query Only)**

Returns the horizontal trigger parameter.
**Group**  Horizontal

**Syntax**  HORizontal:TRIGger?

![Diagram](image)

**Examples**  HORizontal:TRIGger?

**HORizontal:TRIGger:POSition**

Sets or queries the position of the trigger. This is equivalent to setting Trigger Position in the Horizontal menu.

**Group**  Horizontal

**Syntax**  HORizontal:TRIGger:POSition <NR1>

HORizontal:TRIGger:POSition?

![Diagram](image)

**Arguments**  <NR1> is from 0 to 100 % and is the amount of pretrigger information in the waveform.

**Examples**  HORizontal:TRIGGER:POSition?
might return 50.

**ID? (Query Only)**

Returns identifying information about the instrument and its firmware.

**Group**  Status and Error

**Related Commands**  *IDN?
Command Descriptions

**Syntax**

ID?

```
  ID
```

**Returns**

The instrument id in the following format:

```
TEK/<model number>,CF:92.1CT,FV:<firmware version number>
```

**Examples**

ID?
might return TEK/TDS 544A,CF:92.1CT,FV:3.0

---

***IDN? (Query Only)**

Returns the digitizing oscilloscope identification code.

**Group**

Miscellaneous

**Related Commands**

ID

**Syntax**

*IDN?

```
  *IDN
```

**Returns**

The instrument id in the following format:

```
TEKTRONIX,<model number>,0,CF:92.1CT FV:<firmware version number>
```

**Examples**

*IDN?
might return the response
TEKTRONIX,TDS 544A,0,CF:92.1CT FV:2.0

---

**LIMit:BELL**

Rings the bell when the waveform data exceeds the limits set in the limit test, if the limit state is on.

**Group**

Limit Test
**LIMit:COMPARE:CH<x>**

Sets or queries the template against which to compare the waveform acquired through the specified channel. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

---

**Related Commands**

CURve, LIMit:COMPARE:MATH<x>, LIMit:TEMPLate, LIMit:TEMPLate:DESTination, LIMit:TEMPLate:SOUrce, WFMPre

**Syntax**

LIMit:COMPARE:CH<x> { NONe | REF<x> }  
LIMit:COMPARE:CH<x>?

---

**LIMit:BELL**

Sets or queries the template against which to compare the waveform acquired through the specified channel. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

---

**Syntax**

LIMit:BELL { OFF | ON | <NR1> }

LIMit:BELL?

---

**Arguments**

OFF or <NR1> = 0 turns off ringing the bell when any waveform data exceeds the limits set by the limit test.

ON or <NR1> ≠ 0 turns on ringing the bell.

---

**Examples**

LIMit:BELL ON specifies that the bell is to ring when any waveform data exceeds the limits specified in the limit test.

LIMit:BELL?

returns either 0 or 1, indicating whether the bell is to ring when any waveform data exceeds the limits specified in the limit test.
Arguments

REF<x> is a reference waveform.
NONE specifies that no template testing is to be done for the specified channel.

Examples

LIMIT:COMPARE:CH1 REF1
specifies REF1 as the template waveform against which to compare waveforms acquired using CH1.

LIMIT:COMPARE:CH2?
might return LIMIT:COMPARE:CH2 REF4, indicating that waveforms acquired using CH2 will be compared to the template waveform stored in REF4.

LIMIT:COMPARE:MATH<x>
TDS 510A, 500B, 600B, & 700A Only
Sets or queries the template against which to compare a math waveform. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

Group
Limit Test

Related Commands
CURve.LIMIT:TEMPLate, LIMIT:TEMPLate:DESTination, LIMIT:TEMPLate:SOURce, WFMPre

Syntax

LIMIT:COMPARE:MATH<x> { NONE | REF<x> }
LIMIT:COMPARE:MATH<x>?

Arguments

REF<x> is a reference waveform.
NONE specifies that no template testing is to be done for the specified channel.
MATH<x> is a math waveform.

**Examples**

LIMIT:COMPARE:MATH1 REF1
specifies REF1 as the template waveform against which to compare waveforms acquired using MATH1.

LIMIT:COMPARE:MATH2
might return LIMIT:COMPARE:MATH2 REF4, indicating that waveforms acquired using MATH2 will be compared to the template waveform stored in REF4.

**LIMIT:HARDCopy**

Executes a hardcopy operation on the waveform when any waveform data exceeds the limits set in the limit test, if the limit state is on. The hardcopy operation uses the port, and prints in the format and layout, specified using the HARDCopy commands.

**Group**
Limit Test

**Related Commands**
LIMIT:COMPARE:CH<x>, LIMIT:STATE, HARDCopy

**Syntax**
LIMIT:HARDCopy { OFF | ON | <NR1> }

LIMIT:HARDCopy?

**Arguments**

ON or <NR1> ≠ 0 turns on the hardcopy operation for the waveform when any waveform data exceeds the limits set by the limit test.

OFF or <NR1> = 0 turns off the hardcopy operation.

**Examples**

LIMIT:HARDCopy ON
specifies that the hardcopy operation occurs for the waveform when any waveform data exceeds the limits specified in the limit test.
LIMit:HARDCopy?
returns either 0 or 1, indicating whether the hardcopy operation occurs for the waveform when any waveform data exceeds the limits specified in the limit test.

**LIMit:STATE**

Turns limit testing on or off, or queries whether limit testing is in effect.

**Group**  
Limit Test

**Related Commands**  
CURve, LIMit:BELl, LIMit:COMPARE:CH<x>, LIMit:HARDCopy, LIM- it:TEMPLate, WFMPre

**Syntax**  
LIMit:STATE { OFF | ON | <NR1> }

LIMit:STATE?

**Arguments**  
OFF or <NR1> = 0 turns off limit testing.
ON or <NR1> ≠ 0 turns on limit testing.

**Examples**  
LIMit:STATE ON  
specifies that limit testing of waveforms is in effect.
LIMit:STATE?
returns either 0 or 1, indicating whether limit testing of waveforms is in effect.

**LIMit:TEMPLate (No Query Form)**

Creates a template which you can use for limit testing. You can compare the waveform acquired through the specified channel against this template. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

**Group**  
Limit Test
Related Commands

LIM:TEMPLATE:DESTination, LIM:TEMPLATE:SOURce

Syntax

LIM:TEMPLATE STORe

Arguments

STORe creates a template with the specified source waveform and tolerances, and stores it in the destination reference waveform to be used in limit testing comparisons.

Examples

LIM:TEMPLATE STORe
creates a template with the specified source waveform and tolerances, and stores it in the destination reference waveform to be used in limit testing comparisons.

LIM:TEMPLATE:DESTination

Sets or queries the destination reference waveform that the LIM:TEMPLATE STORe command will use.

Group

Limit Test

Related Commands

LIM:COMPARE:CH<x>, LIM:TEMPLATE, LIM:TEMPLATE:SOURce

Syntax

LIM:TEMPLATE:DESTination REF<x>

LIM:TEMPLATE:DESTination?

Arguments

REF<x> specifies the reference waveform destination in which the template waveform is to be stored.

Examples

LIM:TEMPLATE:DESTination REF2
specifies that the template waveform referred to with the LIM:TEMPLATE STORe command is stored as the REF2 waveform.
**LIMit:TEMPLate:SOURce**

Sets or queries the channel, math waveform, or reference waveform that the LIMit:TEMPLate STORe will use.

**Group**  
Limit Test

**Related Commands**  
LIMit:COMPARE:CH<x>, LIMit:TEMPLate, LIMit:TEMPLate:DESTination

**Syntax**  
LIMit:TEMPLate:SOURce { CH<x> | MATH<x> | REF<x> }

LIMit:TEMPLate:SOURce?

**Arguments**  
CH<x> specifies that the template waveform is the waveform currently being acquired using the specified channel.

MATH<x> specifies that the template waveform is the waveform currently stored as the specified math waveform.

REF<x> specifies that the template waveform is the waveform currently stored as the specified reference waveform.

**Examples**  
LIMit:TEMPLate:SOURce CH2
specifies that the template waveform for limit tests is the waveform currently acquired using channel 2.

LIMit:TEMPLate:SOURce?
might return MATH3, specifying that the template waveform for limit tests is the waveform currently stored as the MATH3 waveform.

**LIMit:TEMPLate:TOLerance:HORizontal**

Sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform.

**Group**  
Limit Test
Related Commands  LIMit:COMPARE:CH<x>

Syntax  LIMit:TEMPLate:TOLerance:HOriZontal <NR3>
        LIMit:TEMPLate:TOLerance:HOriZontal?

Arguments  <NR3> is the amount, in horizontal divisions, by which the current waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.

Examples  LIMit:TEMPLate:TOLerance:HOriZontal 1.0 specifies that the current waveform is deemed to be close enough to the template waveform if it is within ±1.0 horizontal division.

          LIMit:TEMPLate:TOLerance:HOriZontal?
          might return 1.0, specifying that the current waveform is close enough to the template waveform if it is within ±1.0 horizontal division.

LIMit:TEMPLate:TOLerance:VERTical

Sets or queries the amount, in units of vertical divisions, by which the source waveform is varied vertically when creating the destination waveform.

Group  Limit Test

Related Commands  LIMit:COMPARE:CH<x>

Syntax  LIMit:TEMPLate:TOLerance:VERTical <NR3>
        LIMit:TEMPLate:TOLerance:VERTical?
Command Descriptions

Arguments

<NR3> is the amount, in vertical divisions, by which the current waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.

Examples

LIMIT:TEMPLATE:TOLerance:VERTical 1.0
specifies that the current waveform is close enough to the template waveform if it is within ±1.0 vertical division from the template waveform.

LIMIT:TEMPLATE:TOLerance:VERTical?
omight return 1.0, specifying that the current waveform is close enough to the template waveform if it is within ±1.0 vertical division.

LOCK

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

Group

Miscellaneous

Related Commands

UNLock, Remote Enable Group, Local Lockout Group

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.
NOTE If the digitizing oscilloscope is in the Remote With Lockout State (RWLS), the LOCk NONe command has no effect. For more information see the ANSI-IEEE Std. 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

**Examples**

LOCK ALL
locks the front panel controls.

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

**LRN? (Query Only)**

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

**Group**

Miscellaneous

**Related Commands**

HEADer, SET?, VERBose

**Syntax**

*LRN?

---

NOTE The *LRN? query 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 digitizing oscilloscope as a command string. The VERBose command can still be used normally to specify whether the returned headers should be abbreviated.

**Examples**

*LRN?

a partial response might look like this:

:ACQUIRE:STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;
REPET 1;STOPAFTER RUNSTOP;:DIAG:LOOP:OPTION ONCE;
COUNT 1;:DIAG:STATE HALT;:HEADER 1;:VERBose 1;
:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;
POSITION1 1.00E-6;POSITION2 9.00E-6;SELECT CURSOR1;
MATH<x>? (Query Only)

Returns the definition for the math waveform specified by <x>.

Group  Vertical
Syntax  MATH<x>?

MATH<x>:DEFINE

Allows the user to define new waveforms using mathematical expressions. This is equivalent to selecting Change Math waveform definition in the Math<x> side menu.

Group  Vertical
Syntax  MATH<x>:DEFINE <QString>
        MATH<x>:DEFINE?

Arguments  <QString> contains the mathematical expression. The expression can include any amount of white space. Expressions can be either single or dual waveform expressions. <src> and <function> elements are case independent.

The format for a single waveform expression is:

        <function>({<source> [, <window>], <scaling>, <phase suppression>])

The format for a dual waveform expression is:

        <source><operator><source>

where:

        <function> ::= INV | DIF | FFT | INT
        ■ INVert (for invert): inverts the defined waveform.
- **Differentiate** (available with Advanced DSP Math only): takes the derivative of the selected waveform.

- **FFT** (available with Advanced DSP Math only): provides an FFT of the selected waveform. It uses the format: “FFT(<source>, <window>, <scaling>, <phase suppression>)” where the window, scaling, and phase suppression arguments in the parentheses are optional. You can specify these arguments in any order.

  - **<source>** refers to a signal channel. Valid choices are: CH1, CH2, CH3, CH4, REF1, REF2, REF3, or REF4.

  (The TDS 410A does not include CH3 and CH4.)

  - **<window>** refers to an FFT window. Valid choices are: RECTangular, HAMming, HANning, or BLackmanharris.

  - **<scaling>** provides vertical scaling. Valid choices are: LOGrms, LINearrms, DEGreesphase, or RADiansphase.

  - **<Phase suppression>** is of the range: –100 dB to 100 dB.

- **Integrate** (available on instruments with the Option 2F Advanced DSP Math only): takes the integral of the selected waveform.

  - **<operator>** ::= { + | – | * | / (not available on TDS 400A) }

  - **<source>** ::= { CH<x> | REF<x> }

**Examples**

```
MATH2:DEFINE "Ch1 + Ch2"
```

adds channel 1 and channel 2, and stores the result in MATH2.

```
MATH1:DEFINE "INV( ref4 )"
```

inverts the waveform stored in reference memory location 4 storing the result in MATH1.

```
MATH1:DEFINE "FFT( CH1 )"
```

takes an FFT on the waveform from channel 1 and stores the result in MATH1.

```
MATH1:DEFINE "FFT( CH1, HAMM, LINEARRMS, 20 )"
```

takes an FFT from channel 1, using the HAMMING algorithm, with linear rms scaling, and 20 dB phase suppression. The result is stored in MATH1.

```
MATH1:DEFINE?
```

might return "Ch2*Ref2" as the expression that defines MATH1.
MATH<x>:NUMAVg

*TDS 510A, 500B, 600B, & 700A Only, some models require Option 2F*

Allows the user to declare at what acquisition number the averaging algorithm will begin exponential averaging. Prior to that acquisition number, the algorithm uses stable averaging. This is equivalent to selecting Average in the Math<x> side menu and entering a value with the general purpose knob or the keypad.

**Group** Vertical

**Syntax**

\[
\text{MATH}\langle x \rangle : \text{NUMAVg} \ <\text{NR1}> \\
\text{MATH}\langle x \rangle : \text{NUMAVg}\ ?
\]

**Arguments**

< NR1 > specifies the number of times to successively average the math waveform before completing an acquisition.

**Examples**

MATH2:NUMAVG 10
Successively averages math waveform 2 by 10 times.

MATH2:NUMAVG?
might return 10 indicating 10 math 2 waveforms are successively averaged before a single acquisition occurs.

MATH<x>:PROCessing

*TDS 510A, 500B, 600B, & 700A Only, some models require Option 2F*

Allows the user to turn on or off averaging for the math waveform specified by <x>. This is equivalent to selecting No Extended Process or Average in the Math<x> side menu.

Math averaging allows the digitizing oscilloscope to successively average any acquisition-related math waveform. This can help reduce noise in a math waveform.
**Group** Vertical

**Syntax**

\[ \text{MATH}<x>:\text{PROCessing} \{ \text{OFF} \mid \text{AVErage} \}\]

\[ \text{MATH}<x>:\text{PROCessing}? \]

**Arguments**

OFF turns off waveform averaging.

AVerage turns on waveform averaging.

**Examples**

\[ \text{MATH1:PROCESSING OFF} \]

ensures that waveform averaging is not in use on math waveform 1.

\[ \text{MATH1:PROCESSING AVERAGE} \]

turns on waveform averaging on math waveform 1.

---

**MEASUrement? (Query Only)**

Returns all measurement parameters.

**Group** Measurement

**Syntax** MEASUrement?

---

**Example** MEASUREMENT?

might return

\[ \text{:MEASUREMENT:MEAS1:TYPE PERIOD;UNITS } "s";\text{SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS};\text{:MEASUREMENT:MEAS2:TYPE PERIOD;UNITS } "s";\text{SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS};\text{:MEASUREMENT:MEAS3:STATE 0};\text{:MEASUREMENT:MEAS3:TYPE PERIOD;UNITS } "s";\text{SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS};\text{:MEASUREMENT:MEAS4:STATE 0};\text{:MEASUREMENT:MEAS4:TYPE PERIOD;UNITS } "s";\text{SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS}} \]
MEASUrement:CLEARSNapshot

Removes the measurement snapshot display.

Group Measurement

Syntax MEASUrement:CLEARSNapshot

Examples MEASUrement:CLEARSNAPSHOT

MEASUrement:GATing

Sets or queries measurement gating.

Group Measurement

Related Commands CURSor:VBAr s

Syntax MEASUrement:GATing { ON | OFF | <NR1> }

MEASUrement:GATing?

Arguments ON (or 1) turns on measurement gating.
OFF (or 0) turns off measurement gating.

**Examples**

```plaintext
MEASUREMENT:GATING ON

MEASUREMENT:GATING?
might return MEASUREMENT:GATING 1
showing gating is turned on.
It might also return MEASUREMENT:GATING 0
showing gating is turned off.
```

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

Returns all immediate measurement setup parameters.

**Group** Measurement

**Syntax** MEASurement:IMMed?

```plaintext
MEASUREMENT:IMMED?
might return :MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS
```

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

Returns information about the immediate delay measurement.

**Group** Measurement

**Syntax** MEASurement:IMMed:DELay?

```plaintext
MEASUREMENT:IMMED:DELAY?
might return :MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS
```
MEASUrement:IMMed:DELay:DIREction

Sets or queries the starting point and direction that determines the delay “to” edge when taking an immediate delay measurement. Use the MEASUrement:IMMed:SOURCE2 command to specify the delay “to” waveform.

Group       Measurement

Syntax       MEASUrement:IMMed:DELay:DIREction {BACKwards | FORwards}  
              MEASUrement:IMMed:DELay:DIREction?

Arguments     BACKwards means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:IMMed:DELay:EDGE1.

FORwards means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:IMMed:DELay:EDGE2.

Examples      MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS
              starts searching from the beginning of the waveform record.

              MEASUREMENT:IMMED:DELAY:DIRECTION?
              returns either BACKWARDS or FORWARDS.

MEASUrement:IMMed:DELay:EDGE1

Sets or queries the slope of the edge that is used for the delay “from” waveform when taking an immediate delay measurement. The waveform is specified by MEASUrement:IMMed:SOURCE1.

Group       Measurement
**Related Commands**

`MEASUrement:IMMed:SOURCE1`

**Syntax**

`MEASUrement:IMMed:DELay:EDGE1 { FALL | RISe }`

`MEASUrement:IMMed:DELay:EDGE1?`

**Arguments**

FALL specifies the falling edge.

RISe specifies the rising edge.

**Examples**

`MEASUREMENT:IMMED:DELAY:EDGE1 RISE`

specifies that the rising edge be used for the immediate delay measurement.

`MEASUREMENT:IMMED:DELAY:EDGE1?`

returns either RISE or FALL.

---

**MEASUrement:IMMed:DELay:EDGE2**

Sets or queries the slope of the edge that is used for the delay “to” waveform when taking an immediate delay measurement. The waveform is specified by `MEASUrement:IMMed:SOURCE2`.

**Group**

Measurement

**Related Commands**

`MEASUrement:IMMed:SOURCE2`

**Syntax**

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

`MEASUrement:IMMed:DELay:EDGE2?`
Arguments  
FALL specifies the falling edge.
RISe specifies the rising edge.

Examples  
MEASUREMENT:IMMed:DELAY:EDGE2 RISE
specifies that the rising edge be used for the immediate delay measurement.
MEASUREMENT:IMMed:DELAY:EDGE2?
returns FALL showing that the falling or negative edge of the waveform is used for the immediate delay measurement.

**MEASUrement:IMMed:SOURCE[1]**

Sets or queries the source for all single channel immediate measurements and specifies the source to measure “from” when taking an immediate delay measurement or phase measurement.

**Group**  
Measurement

**Syntax**  
MEASUREMENT:IMMed:SOURCE[1] { CH<x> | MATH<x> | REF<x> }
MEASUREMENT:IMMed:SOURCE[1]?

Arguments  
CH<x> is an input channel.
MATH<x> is a math waveform.
REF<\text{x}> is a reference waveform.

**Examples**

\text{MEASUREMENT:IMMed:SOURCE MATH1}

specifies MATH1 as the immediate measurement source.

**MEASUREMENT:IMMed:SOURCE2**

Specifies the source to measure “to” when taking an immediate delay measurement or phase measurement.

**Group** Measurement

**Syntax**

\text{MEASUREMENT:IMMed:SOURCE2 \{ CH<\text{x}> | MATH<\text{x}> | REF<\text{x}> \}}

\text{MEASUREMENT:IMMed:SOURCE2?}

**Arguments**

CH<\text{x}> is an input channel.

MATH<\text{x}> is a math waveform.

REF<\text{x}> is a reference waveform.

**Examples**

\text{MEASUREMENT:IMMed:SOURCE2 REF3}

sets the waveform in reference memory location 3 as the delay “to” source when making delay measurements.

\text{MEASUREMENT:IMMed:SOURCE2?}

might return MATH1.

**MEASUREMENT:IMMed:TYPE**

Specifies the immediate measurement.

**Group** Measurement
Syntax

MEASurement:IMMed:TYPE { AMplitude | AREa | BURst | CARea | CMEan |
| CRMs | DELay | FALL | FREQuency | HIGH | LOW | MAXimum | MEAN |
| MINimum | NDUty | NOvershoot | NWIdth | PDUty | PERiod | PHAse |
| PK2pk | POvershoot | PWIdth | RISE | RMS }

MEASurement:IMMed:TYPE?

Arguments

For more information on how measurements are derived, see Appendix B: Algorithms, in the related TDS Family User Manual.
**AMP1tude** is the high value minus the low value.

**AREA** is the area between the curve and ground over the entire waveform.

**BURSt** is the time from the first MidRef crossing to the last MidRef crossing.

**CARRea** (cycle area) is the area between the curve and ground over one cycle.

**CMEAn** is the arithmetic mean over one cycle.

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

**DELay** is the time between the MidRef crossings of two different waveforms.

**FALL** is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value of its final value.

**FREQuency** is the reciprocal of the period measured in hertz.

**HIGH** is the 100% reference level.

**LOW** is the 0% reference level.

**MAXimum** is the highest amplitude (voltage).

**MEAN** is the arithmetic mean over the entire waveform.

**MINimum** is the lowest amplitude (voltage).

**NDUty** is the ratio of the negative pulse width to the signal period expressed as a percentage.

**NOVershoot** is the negative overshoot, expressed as:

\[ NOVershoot = 100 \times \left( \frac{(Low - Minimum)}{Amplitude} \right) \]

**NWIdth** is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

**PDUty** is the ratio of the positive pulse width to the signal period expressed as a percentage.

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

**PHAs e** is the amount of shift, expressed in degrees, from the selected waveform to the designated waveform.

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

**POVershoot** is the positive overshoot, expressed as:
POVershoot = 100 \times \left( \frac{(Maximum - High)}{Amplitude} \right)

**POVershoot** is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

**RWidth** is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

**RISe** is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value of its final value.

**RMS** is the true Root Mean Square voltage.

**Examples**

MEASUREMENT:IMMed:TYPE FREQUENCY
defines the immediate measurement to be a frequency measurement.

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

Returns the units for the immediate measurement.

**Group**

Measurement

**Related Commands**

MEASUrement:IMMed:TYPE

**Syntax**

MEASUREMENT:IMMed:UNITS?

**Returns**

<QString> returns "V" for volts, "s" for seconds, "Hz" for hertz, "V^2" for volts^2 or "%" for percent.

On the TDS 400A, <QString> also returns "c" for clocks, "Vc" for volt-clocks, or "/c" for 1/clks.

**Examples**

MEASUREMENT:IMMed:UNITS?

might return "s", indicating that the units for the immediate measurement are seconds.
MEASUrement:IMMed:VALue? (Query Only)

Immediately executes the immediate measurement specified by the MEASUrement:IMMed:TYPe command. The measurement is taken on the source(s) specified by a MEASUrement:IMMed:SOURCe command.

- **Group**: Measurement
- **Syntax**: MEASUrement:IMMed:VALue?
- **Returns**: <NR3>

MEASUrement:MEAS<x>? (Query Only)

Returns all measurement parameters for the displayed measurement specified by <x>.

- **Group**: Measurement
- **Syntax**: MEASUrement:MEAS<x>?
- **Examples**: MEASUREMENT:MEAS3? might return :MEASUREMENT:MEAS3:STATE 0;TYPE PERIOD; UNITS "s";SOURCE1 CH1;SOURCE2 CH2;DELAY:EDGE1 RISE; EDGE2 RISE;DIRECTION FORWARDS.

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

Returns the delay measurement parameters for the measurement specified by <x>.

- **Group**: Measurement
- **Syntax**: MEASUrement:MEAS<x>:DELay?
MEASurement:MEAS<x>:DELay:DIREction

Sets or queries the starting point and direction that determines the delay “to” edge when taking a delay measurement. The waveform is specified by MEASurement:MEAS<x>:SOURCE2. This command is equivalent to setting the direction in the Delay Edges & Direction side menu.

**Group**  Measurement

**Syntax**  MEASurement:MEAS<x>:DELay:DIREction { BACKwards | FORwords }

MEASurement:MEAS<x>:DELay:DIREction?

**Arguments**  BACKwards means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. The slope of the edge is specified by MEASurement:MEAS<x>:DELay:EDGE2.

FORwards means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. The slope of the edge is specified by MEASurement:MEAS<x>:DELay:EDGE2.

**Examples**  MEASUREMENT:MEAS1:DELAY:DIRECTION BACKWARDS

starts searching from the end of the waveform record.

MEASUREMENT:MEAS3:DELAY:DIRECTION?

might return FORWARDS for the search direction.
**MEASUrement:MEAS<x>:DELay:EDGE1**

Sets or queries the slope of the edge that is used for the delay “from” waveform when taking a delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOURCE1. This command is equivalent to selecting the edges in the Delay Edges & Direction side menu.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:DELay:EDGE1 { FALL | RISE }  
MEASUrement:MEAS<x>:DELay:EDGE1?

**Arguments**  
FALL specifies the falling edge.  
RISe specifies the rising edge.

**Examples**  
MEASUREMENT:MEAS3:DELAY:EDGE1 RISE  
specifies that the rising edge be used for measurement 3.  
MEASUREMENT:MEAS1:DELAY:EDGE1?  
returns either RISE or FALL for measurement 1.

**MEASUrement:MEAS<x>:DELay:EDGE2**

Sets or queries the slope of the edge that is used for the delay “to” waveform when taking a delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOURCE2. This command is equivalent to selecting the edges in the Delay Edges & Direction side menu.

**Group**  
Measurement

**Syntax**  
MEASUrement:MEAS<x>:DELay:EDGE2 { FALL | RISE }
**MEASurement:MEAS<x>:DELy:EDGE2?**

Arguments

- FALL specifies the falling edge.
- RISE specifies the rising edge.

Examples

- `MEASUREMENT:MEAS2:DELAY:EDGE2 RISE` specifies that the rising edge be used for the second delay measurement.
- `MEASUREMENT:MEAS2:DELAY:EDGE2?` might return FALL showing that the falling or negative edge of the waveform is used for the second measurement.

**MEASurement:MEAS<x>:SOURCE[1]**

Sets or queries the source for all single channel measurements and specifies the source to measure “from” when taking a delay measurement or phase measurement.

Group

Measurement

Syntax

```
MEASUREMENT:MEAS<x>:SOURCE[1] { CH<x> | MATH<x> | REF<x> }
MEASUREMENT:MEAS<x>:SOURCE[1]?
```

Arguments

- CH<x> is an input channel.
is a math waveform.

REF<x> is a reference waveform.

**Examples**

MEASUREMENT:MEAS2:SOURCEx MATH1
specifies MATH1 as the measurement 2 source.

**MEASUrement:MEAS<x>:SOURCE2**

Sets or queries the source to measure “to” when taking a delay measurement or phase measurement. This is equivalent to setting the source in the Delay from Selected Wfm side menu or the Phase from Selected Wfm side menu.

**Group**
Measurement

**Syntax**

```
MEASUrement:MEAS<x>:SOURCE2 { CH<x> | MATH<x> | REF<x> }
MEASUrement:MEAS<x>:SOURCE2?
```

**Arguments**

CH<x> is an input channel.

MATH<x> is a math waveform.

REF<x> is a reference waveform.

**Examples**

MEASUREMENT:MEAS4:SOURCEx CH<x>
sets channel 1 as the delay “to” source when making delay measurements.

MEASUREMENT:MEAS2:SOURCEx?
might return MATH1.
MEASUrement:MEAS<x>:STATE

Controls the measurement system. The source specified by MEASUrement:MEAS<x>:SOURCE1 must be selected for the measurement to be displayed. The source can be selected using the SELect:CH<x> command.

Group Measurement

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

MEASUrement:MEAS<x>:STATE?

Arguments OFF or <NR1> = 0 turns measurements off. You can also turn the state off by deselecting the source.

ON or <NR1> ≠ 0 turns measurements on.

Examples MEASUrement:MEAS1:STATE ON

turns measurement defined as MEAS1 on.

MEASUrement:MEAS4:STATE?

returns either 0 or 1, indicating the state of MEAS4.

MEASUrement:MEAS<x>:TYPE

Sets or queries the measurement type for the measurement specified by MEAS<x>. This is equivalent to selecting the measurement in the Select Measurement side menu.

Group Measurement

Syntax MEASUrement:MEAS<x>:TYPE { AMPlitude | AREa | BURst | CAREa |

| CMean | CRMs | DELay | FALL | FREQuency | HIGH | LOW | MAXimum | MEAN | MINimum | NDUty | NOVershoot | NWIdth | PDUty | PERIod |

| PHAse | PK2pk | POVershoot | PWIdth | RISE | RMS }
MEASUREment:MEAS<x>:TYPE?

**Arguments**

For more information on how measurements are derived, see Appendix B: Algorithms, in the related TDS Family User Manual.

AMP1tude is the high value minus the low value or HIGH – LOW.

AREa is the area between the curve and ground over the entire waveform.

BURSt is the time from the first MidRef crossing to the last MidRef crossing.
CAREa (cycle area) is the area between the curve and ground over one cycle.

CMean is the arithmetic mean over one cycle.

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

DELay is the time between the MidRef crossings of two different waveforms.

FALL is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value of its final value.

FREQuency is the reciprocal of the period measured in hertz.

HIGH is the 100% reference level.

LOW is the 0% reference level.

MAXimum is the highest amplitude (voltage).

MEAN is the arithmetic mean over the entire waveform.

MINimum is the lowest amplitude (voltage).

NDuty is the ratio of the negative pulse width to the signal period expressed as a percentage.

NOVershoot is the negative overshoot, expressed as:

$$NOVershoot = 100 \times \left( \frac{(Low - Minimum)}{Amplitude} \right)$$

NWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

PDuty is the ratio of the positive pulse width to the signal period expressed as a percentage.

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

PHASE is the amount of shift, expressed in degrees, from the selected waveform to the designated waveform.

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

POVershoot is the positive overshoot, expressed as:

$$POVershoot = 100 \times \left( \frac{(Maximum - High)}{Amplitude} \right)$$
PWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

RISe is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value of its final value.

RMS is the true Root Mean Square voltage.

**Examples**

```
MEASUREMENT:MEAS3:TYPE RMS
```
specifies MEAS3 to calculate the Root Mean Square voltage.

---

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

Returns the units for the measurement specified by MEASUrement:MEAS<x>:TYPe.

**Group** Measurement

**Syntax**

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

**Returns**

\(<\text{Q}String>\) returns "V" for volts, "s" for seconds, "HZ" for hertz, "VW" for volts², or "p" for percent.

On the TDS 400A, \(<\text{Q}String>\) also returns "c" for clocks, "Vc" for volt-clocks, or "c" for 1/clks.

**Examples**

```
MEASUREMENT:MEAS3:UNITS?
```
might return "p", indicating the units for Measurement 3 are percent.

---

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

Returns the value that has been calculated for the measurement specified by \(<x>\).

**NOTE** This value is a display value and will be updated perhaps every 1/3 second. If you are acquiring a long acquisition record, the TDS may take longer than this time to update.
MEASUrement:METHod

Sets or queries the method used to calculate the 0% and 100% reference level. This is equivalent to setting the **High-Low Setup** in the Measure menu.

**Group**  Measurement

**Syntax**  MEASUrement:METHod \{ HISTogram | MINMax \}

**Arguments**

- **HISTogram** sets the high and low waveform levels statistically using a histogram algorithm.
- **MINMax** sets the high and low waveform levels to MAX and MIN, respectively.

**Examples**

- `MEASUREMENT:METHOD HISTOGRAM` specifies that the high and low reference levels are set statistically.
- `MEASUREMENT:METHod?` returns MINMAX when the reference levels are set to MIN and MAX.

MEASUrement:REFLevel? (Query Only)

Returns the reference levels.

**Group**  Measurement
MEASUrement:REFLevel:ABSolute:HIGH

Sets or queries the high reference level, and is the 100% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. This command is equivalent to setting the Reference Levels in the Measure menu.

Syntax

```
MEASUrement:REFLevel?
```

Arguments

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

Examples

```
MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.71
```
sets the high reference level to 1.71 V.

MEASUrement:REFLevel:ABSolute:LOW

Sets or queries the low reference level, and is the 0% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. This command is equivalent to setting the Reference Levels in the Measure menu.

Syntax

```
MEASUrement:REFLevel:ABSolute?
```

Arguments

<NR3> is the low reference level, in volts. The default is 0.0 V.
Arguments  <NR3> is the low reference level, in volts. The default is 0.0 V.

Examples  MEASUREMENT:REFLEVEL:ABSOLUTE:LOW?
          might return 0.0E+0 as the low reference level.

**MEASUrement:REFLevel:ABSolute:MID**

Sets or queries the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. This command is equivalent to setting the **Reference Levels** in the Measure menu.

**Group**  Measurement

**Syntax**  MEASUREMENT:REFLEVEL:ABSOLUTE:MID <NR3>

            MEASUREMENT:REFLEVEL:ABSOLUTE:MID?

Arguments  <NR3> is the mid reference level, in volts. The default is 0.0 V.

Examples  MEASUREMENT:REFLEVEL:ABSOLUTE:MID .71
          sets the mid reference level to .71 volts.
**MEASUrement:REFLevel:ABSolute:MID2**

Sets or queries the mid reference level for the “to” waveform when taking a delay measurement, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. This command is equivalent to setting the Reference Levels in the Measure menu.

**Group** Measurement

**Syntax**

```
MEASUrement:REFLevel:ABSolute:MID2 <NR3>
MEASUrement:REFLevel:ABSolute:MID2?
```

**Arguments**

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

**Examples**

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5
```

sets the mid reference level for the delay waveform to 0.5 volts.

**MEASUrement:REFLevel:METHod**

Specifies which reference levels are used for measurement calculations. This command is equivalent to setting the levels in the Reference Levels side menu.

**Group** Measurement

**Syntax**

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

sets the reference levels for measurement calculations.
Arguments

**ABSolute** specifies that the reference levels are set explicitly using the `MEASUrement:REFLevel:ABSolute` commands. This method is useful when precise values are required. For instance, when designing to published interface specifications such as RS-232-C.

**PERCent** specifies that the reference levels are calculated as a percent relative to **HIGH** and **LOW**. The percentages are defined using the `MEASUrement:REFLevel:PERCent` commands.

Examples

`MEASUrement:REFLevel:METHOD ABSolute` specifies that explicit user-defined values are used for the reference levels.

`MEASUrement:REFLevel:METHOD?` returns either **ABSolute** or **PERCENT**, indicating the reference levels used.

**MEASUrement:REFLevel:PERCent:HIGH**

Sets or queries the percent, where 100% is equal to **HIGH**, that is used to calculate the high reference level when `MEASUrement:REFLevel:METHod` is set to **PERCent**. This command is equivalent to setting the **Reference Levels** in the Measure menu.

Group

Measurement

Syntax

`MEASUrement:REFLevel:PERCent:HIGH <NR3>`

`MEASUrement:REFLevel:PERCent:HIGH?`

Arguments

`<NR3>` ranges from 0 to 100 percent, and is the high reference level. The default is 90%.

Examples

`MEASUrement:REFLevel:PERCent:HIGH 95` specifies that the high reference level is set to 95% of **HIGH**.
MEASUrement::REFLevel::PERCent::LOW

Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the low reference level when MEASUrement::REFLevel::METHod is set to PERCent. This command is equivalent to setting the Reference Levels in the Measure menu.

**Group**  Measurement

**Syntax**  MEASUrement::REFLevel::PERCent::LOW <NR3>

MEASUrement::REFLevel::PERCent::LOW?

**Arguments**  <NR3> ranges from 0 to 100 percent, and is the low reference level. The default is 10%.

**Examples**  MEASUrement::REFLevel::PERCent::LOW?

might return 15, meaning that the low reference level is 15% of HIGH.

MEASUrement::REFLevel::PERCent::MID

Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the mid reference level when MEASUrement::REFLevel::METHod is set to PERCent. This command is equivalent to setting the Reference Levels in the Measure menu.

**Group**  Measurement

**Syntax**  MEASUrement::REFLevel::PERCent::MID <NR3>

MEASUrement::REFLevel::PERCent::MID?
MEASurement:REFLevel:PERCent:MID2

Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the mid reference level for the second waveform specified when taking a delay measurement. This command is equivalent to setting the Reference Levels in the Measure menu.

Group    Measurement

Syntax    MEASurement:REFLevel:PERCent:MID2 <NR3>
          MEASurement:REFLevel:PERCent:MID2?

Arguments <NR3> ranges from 0 to 100 percent, and is the mid reference level. The default is 50%.

Examples  MEASUREMENT:REFLEVEL:PERCENT:MID 60 specifies that the mid reference level is set to 60% of HIGH.

          MEASUREMENT:REFLEVEL:PERCENT:MID2 40 specifies that the mid reference level is set to 40% of HIGH.
**MEASUrement:SNAPShot**

Displays the measurement snapshot.

**Group**  
Measurement

**Syntax**  
MEASUrement:SNAPShot

**Examples**  
MEASUREMENT:SNAPSHOT

**MESSage**

Clears the message window and the MESSage? query returns the current message parameters.

**Group**  
Display

**Syntax**  
MESSage CLEar  
MESSage?

**Arguments**  
CLEar removes the message from the message window. This is equivalent to sending MESSage SHOW "".

**Examples**  
MESSAGE CLEAR  
Clears the message from the window.

**MESSage:BOX**

Defines the size and position of the message window. This command does not display the window unless MESSage:STATE is ON.

**Group**  
Display
**Syntax**

MESSage:BOX <X1>,<Y1>,<X2>,<Y2>
MESSage:BOX?

**Arguments**

<X1> and <X2> = 0 to 640, and are pixel positions along the horizontal axis. <X1> defines the left and <X2> defines the right side of the window.

<Y1> and <Y2> = 0 to 480, and are pixel positions along the vertical axis. <Y1> defines the top and <Y2> defines the bottom of the window. The reserved height of all characters is 15 pixels so the window must be at least that high to fully display characters. For a complete list of character widths in pixels, see Table A–1 on page A–1. Shorter windows clip characters.

Figure 3-1 shows the coordinate system relative to the screen.

![Figure 2–4: Message Window Coordinates](image-url)
MESSage:SHOW

Clears the contents of the message window and displays the new message in the window.

**Group**
Display

**Syntax**
MESSage:SHOW <Qstring>
MESSage:SHOW?

**Arguments**

<Qstring> is the message and can include any of the characters shown in the TDS Character Chart in Appendix A. The maximum length of the message is 1000 characters.

The message is left-justified, and is displayed on a single line starting with the top most line in the window. A line feed character can be embedded in the string to position the message on multiple lines. You can also use white space and tab characters to position the message within a line.

You can send a tab by transmitting a tab character (decimal 9) followed by two characters representing the most significant eight bits followed by the least significant eight bits of a 16-bit number. The number specifies the pixel column relative to the left margin of the label area. For example, to tab to pixel 13, send TAB (decimal 9), NUL (decimal 0), and CR (decimal 13).

The ESC character followed by the @ character turns inverse video on or off and can be embedded in the message string. The first ESC character followed by a @ character displays all the text that follows in inverse video until another ESC character followed by a @ character is found in the string.

**NOTE.** The use of any escape codes other than those described above may produce unpredictable results.

The label area is the height and width you have set using the MESSage:Box command. The length of the label that fits in the label area depends on the contents of the label because the width of characters varies. For a complete list of character widths in pixels, see Table A–1 on page A–1.

If the message exceeds the limits of the window, either horizontally or vertically, the portion of the message that exceeds the limits will not be displayed. The
message string itself is not altered. The entire message can be returned as a query response regardless of what is displayed in the window.

**Examples**

MESSAGE:SHOW "Hello world"
displays “Hello world” in the upper left corner of the box (you can define the box size with the MESSAGE BOX command).

MESSAGE:SHOW "Hello world\# ... hello"
displays “Hello world ... hello” in the upper left corner of the box and the word “world” is displayed in inverse video. In this example, \# stands for the escape character. The escape character may appear differently for you depending on your GPIB talker-listener program.

**MESSage:STATE**

Controls the display of the message window.

**Group**

Display

**Syntax**

MESSage:STATE { OFF | ON | <NR1> }

MESSage:STATE?

**Arguments**

\(<OFF>\) or \(<NR1> = 0\) removes the message window from the screen.

\(<ON>\) or \(\not= 0\) displays the message window and its contents on the screen. The size of the window is defined by MESSage:BOX.

**NEWpass (No Query Form)**

Changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

**Group**

Miscellaneous
Related Commands  PASSWord, *PUD

Syntax  NEWpass <QString>

Arguments  <QString> is the new password. The password can include up to 10 characters.

Examples  NEWPASS "mypassword"
creates a new password for accessing the user protected data.

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

Group  Status and Error

Related Commands  BUSY?, *WAI

Syntax  *OPC
*OPC?

The *OPC command allows you to synchronize the operation of the digitizing oscilloscope with your application program. Synchronization methods are described starting on page 3–7. Table 2–30 shows commands that generate an Operation Complete message.
Table 2–30: Commands that Generate an Operation Complete Message

<table>
<thead>
<tr>
<th>Operation</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic scope adjustment</td>
<td>AUTOSet EXECute</td>
</tr>
<tr>
<td>Internal self-calibration</td>
<td>*CAL</td>
</tr>
<tr>
<td>Single sequence acquisition</td>
<td>ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQuence)</td>
</tr>
<tr>
<td>Hardcopy output</td>
<td>HARDCopy STARt</td>
</tr>
</tbody>
</table>

**OPT? (Query Only)**

*Not on TDS 400A*

Returns a list of options installed in your digitizing oscilloscope.

**Group**

Status and Error

**Syntax**

*OPT?

**Examples**

OPT?


**PASSWord (No Query Form)**

Enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the digitizing oscilloscope is powered off, or until the FACTory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

**Group**

Miscellaneous
**NEWpass, *PUD**

**Syntax**

PASSWord

PASSWord <QString>

**Arguments**

<QString> is the password and can include up to 10 characters. The factory default password is “XYZZY” and is always valid.

**Examples**

PASSWOrd "XYZZY"
Enables the *PUB and NEWpass set commands.

PASSWOrd
Disables the *PUD and NEWpass set commands. You can still use the query version of *PUD.

**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

**Related Commands**

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

**Syntax**

*PSC <NR1>

*PSC?
Arguments

\(<\text{NR1}\> = 0 \) sets the power-on status clear flag to false, disables the power-on clear and allows the digitizing oscilloscope to possibly assert SRQ after power-on.

\(<\text{NR1}\> \neq 0 \) sets the power-on status clear flag true. Sending \(*\text{PSC} \; 1\) therefore enables the power-on status clear and prevents any SRQ assertion after power-on. Using an out-of-range value causes an execution warning.

Examples

\(*\text{PSC} \; 0\) 
sets the power-on status clear flag to false.

\(*\text{PSC}\)?
might return the value 1, showing that the power-on status clear flag is set to true.

\*PUD

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

Group

Miscellaneous

Related Commands

PASSWord

Syntax

\(*\text{PUD} \; <\text{Block}>\)

\(*\text{PUD}?\)

Arguments

\(<\text{Block}>\) is a string containing up to 100 characters.

Examples

\(*\text{PUD} \; #229\)This instrument belongs to me 
stores the string “This instrument belongs to me” in the user protected data area.

\(*\text{PUD}?\)
might return #221Property of Company X.
**RCL (No Query Form)**

Restores the state of the digitizing 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

**Related Commands**  

**Syntax**  
*RCL <NR1>

**Arguments**  
<NR1> is a value in the range from 1 to 10, and specifies a setup storage location. Using an out-of-range value causes an execution error (222, “Data out of range”).

**Examples**  
*RCL 3  
restores the digitizing oscilloscope from a copy of the settings stored in memory location 3.

**RECALL:SETUp (No Query Form)**

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

**Group**  
Save and Recall

**Related Commands**  

**Syntax**  
RECALL:SETUp { FACtory | <NR1> | <file path> }

---

TDS Family Oscilloscope Programmer Manual  
2–205
**Arguments**

FACtory selects the factory setup.

<NR1> is a value in the range from 1 to 10 and specifies a setup storage location. Using an out-of-range value causes an execution error (222, “Data out of range”).

<file path> (available on instruments with the Option 1F File System) is the location in mass storage memory where the setup will be recalled from.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will read the file from the default directory. <filename> stands for a filename of up to 8 characters followed by a period ("." ) and any 3-character extension. Do not use wild card characters.

**Examples**

RECALL:SETUP FACTORY
recalls (and makes current) the front panel setup to its factory defaults.

RECALL:SETUP 1
recalls the front panel setup from setup1.

RECALL:SETUP "TEK00000.SET"
recalls the front panel setup from the file TEK00000.SET in the default directory and on the default drive.

**RECALL:WAVEform (No Query Form)**

*File System Only*

Recalls a stored waveform into a reference location. This command is equivalent to selecting Recall File in the Save/Recall Waveform menu.

**Group**

Save and Recall

**Syntax**

RECALL:WAVEform <file path>,REF<x>

**Arguments**

REF<x> is the location in internal reference memory where the waveform is recalled from.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> “and one or more <dir>s are optional. If you do not specify them, the TDS will recall the waveform from the default directory. <filename> stands for a filename of up to
8 characters followed by a period (’.’) and any 3-character extension. Do not use wild card characters.

**Examples**

RECALL:WAVEFORM "TEK00000.WFM",REF1
recalls the waveform stored in the file named TEK00000.WFM to reference location 1.

**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.

**RST (No Query Form)**

(Reset) returns the digitizing oscilloscope to a known set of instrument settings, but does not purge any aliases or stored settings.

**Group**  
Status and Error

**Related Commands**  

**Syntax**  
*RST

*RST returns the instrument settings to the factory defaults (see Appendix D: Factory Initialization Settings).
The *RST command does not alter the following:

- The state of the IEEE Std 488.1–1987 interface.
- The selected IEEE Std 488.1–1987 address of the digitizing oscilloscope.
- Calibration data that affect device specifications.
- The Output Queue.
- The Service Request Enable Register setting.
- The Standard Event Status Enable Register setting.
- The Power-on status clear flag setting.
- Alias definitions.
- Stored settings.
- The *PUD? response.

**RS232:BAUd**

*RS-232/Centronics Hardcopy Interface Only*

Sets or queries RS-232-C interface transmission speed.

**Group**

RS232

**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.
**RS232:HARDFlagging**  
*RS-232/Centronics Hardcopy Interface Only*

Sets or queries the input and output hard flagging over the RS-232 port. It uses the RFR (Ready For Receive) and CTS (Clear To Send) lines to control data transmission. On output, the oscilloscope transmits data only when CTS is asserted. When CTS is not asserted, the oscilloscope stops transmitting data. On input, it asserts RFR until the receive queue is full. Then it unasserts RFR to stop transmission from an external printer. CTS remains unasserted until the receive queue is not full. At that time, CTS is asserted again to restart transmission.

**Group**  
RS232

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

**Examples**  
RS232:HARDFlagging ON  
turns on hard flagging.

**RS232:PARity**  
*RS-232/Centronics Hardcopy Interface Only*

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

**Group**  
RS232
Syntax  RS232:PARity \{ EVEN | ODD | NONE \}

RS232:PARity?

Arguments  EVEN indicates the parity bit is sent with even parity and bytes received are expected to have even parity.

ODD indicates the parity bit is sent with odd parity and bytes received are expected to have odd parity.

NONE indicates that no parity bit is sent and none are expected.

Examples  RS232:PARITY EVEN

sets the parity to even.

**RS232:SOFTFlagging**

*RS-232/Centronics Hardcopy Interface Only*

Sets or queries the input and output soft flagging over the RS-232 port. It stops transmitting data any time it receives an XOFF (DC3) character. It sends an XOFF character when its 512 byte input buffer has 80 free bytes. The digitizing oscilloscope begins transmitting data again when it receives an XON (DC1) character. It sends XON when its input buffer has 100 free bytes.

Group  RS232

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

RS232:SOFTFlagging?
Arguments

ON or <NR1> \neq 0 turn on softflagging.

OFF or <NR1> = 0 turn off softflagging.

Examples

RS232:SOFTFLAGGING ON

turns on soft flagging.

RS232:STOPBits

RS-232/Centronics Hardcopy Interface Only

Sets or queries the number of transmission stop bits sent with each character to identify the end of data for that character.

Group

RS232

Syntax

RS232:STOPBits <NR1>

RS232:STOPBits?

Arguments

<NR1> is 1 or 2.

Examples

RS232:STOPBITS 1 sets the number of stop bits to 1.

RS232? (Query Only)

RS-232/Centronics Hardcopy Interface Only

Queries the RS232 settings.

Group

RS232

Syntax

RS232?
Arguments  None


It might return:

RS232 BAUD: 9600, SOFTFLAGGING: OFF, HARDFLAGGING: ON, PARITY: NONE, STOPBITS: 1

**SAV (No Query Form)**

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

Group  Save and Recall

Related Commands  DELEte:SETUp, FACtory, *RCL, RECALL:SETUp, SAVe:SETUp

Syntax  **SAV** \(<NR1>\)

Arguments  \(<NR1>\) is a value in the range from 1 to 10 and specifies a location. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

Examples  *SAV 2

saves the current settings in memory location 2.

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

Saves the current front-panel setup into the specified memory location or file. This is equivalent to selecting the Save Current Setup in the Save/Recall Setup menu.

Group  Save and Recall
**SAVE:SETUp (No Query Form)**

Stores a waveform in one of four reference memory locations or a mass storage file (on instruments with the Option 1F File System). This command is equivalent to selecting either the Save Waveform or the Save to File Waveform item in the Save/Recall Waveform menu.

**Group**  
Save and Recall

**Related Commands**  
DELEte:WAVEform, SAVE:WAVEform:FILEFormat

**Syntax**  
SAVE:SETUp { <NR1> | <file path> }

**Arguments**  
*NR1* is a value in the range from 1 to 10 and specifies a location. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

*file path* (available on instruments with the Option 1F File System) is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>.<drive> and one or more <dir>s are optional. If you do not specify them, the TDS will write the file to the default directory. <filename> stands for a filename of up to 8 characters followed by a period ("."), and the 3-char extension “SET”. The TDS will generate an error if you use any other extension for saving a setup.

Settings saved in one TDS oscilloscope may or may not work on a different model TDS or on the same model TDS with a different version of firmware.

**Examples**  
SAVE:SETUP 5  
saves the current front-panel setup in memory location 5.

SAVE:SETUP "TEK00000.SET"  
saves the current front-panel setup to the file TEK00000.SET in the default directory and on the default drive.
Arguments

<wfm> is CH<x>, MATH<x>, or REF<x>, and is the waveform that will be saved.

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

<file path> (on instruments with the Option 1F File System) is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the digitizing oscilloscope will write the file to the default directory. <filename> stands for a filename of up to 8 characters followed by a period ("." ) and the proper 3-character extension. Internal format waveforms use the “WFM” extension. The TDS 400A and TDS 700A can also use a “CSV” extension for spreadsheet format files or a “DAT” extension for MathCad format files. The digitizing oscilloscope will generate an error if you use any other extension for saving a waveform.

Examples

SAVE:WAVEFORM MATH2,REF1
saves the math 2 waveform in reference memory location 2.

SAVE:WAVEFORM MATH1,"TEK00000.WFM"
saves the math1 waveform to the file TEK00000.WFM in the default directory and on the default drive.

SAVE:WAVEform:FILEFormat
Not on TDS 510A

Specifies the file format for saved waveforms.

Group
Save and Recall

Related Commands
SAVE:WAVEform

Syntax
SAVE:WAVEform:FILEFormat{ INTERNAL | SPREADsheet | MATHcad }
SAVE:WAVEform:FILEFormat?
Arguments

- **INTERNAL** specifies the internal format. Internal format files have a .wfm extension.
- **SPREADSheet** specifies the spreadsheet format. Spreadsheet format files have a .CSV extension.
- **MATHCad** specifies the MathCad format. MathCad format files have a .DAT extension.

If you are writing a MathCad program, you should be aware that the TDS-MathCad file has the following features:

- ASCII format
- first four values contain header information
- first header value holds the TDS record length
- second header value holds time, in seconds, between samples
- third header value holds the trigger position (expressed as an index in the data position)
- fourth header value refers to the fractional trigger position
- delimiters are carriage returns

Examples

```
SAVE:WAVEFORM:FILEFORMAT SPREADSHEET
```

specifies the waveform, when saved, will be stored in a spreadsheet–compatible format.

**SELect? (Query Only)**

Returns the selected waveform and the display status of all waveforms.

**Group** Vertical

**Syntax** SELect?
**SELect:**<wfm>

Controls the display and selection of waveforms. There can be up to eleven waveforms displayed at one time, but only one waveform can be selected at a time. The selected waveform is the waveform that was most recently turned on. This command is equivalent to pressing a front-panel CH or MORE button. <wfm> can be CH<x>, MATH<x>, or REF<x>.

**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> ≠ 0 turns on the display of the specified waveform. The waveform also becomes the selected waveform.

**Examples**  
SELECT:CH2 ON  
turns the channel 2 display on and selects channel 2.

SELECT:REF1?
returns either 0 or 1, indicating whether the REF1 waveform is selected.
SELect:CONTROI

Sets or queries the waveform that is currently affected by the cursor and vertical commands.

**Group**  Vertical

**Syntax**  SELect:CONTROI <wfm>
             SELect:CONTROI?

**Arguments**  <wfm> is CH<x>, MATH<x>, or REF<x>, and is the selected waveform.

**Examples**  SELect:CONTROI?
               might return CH1 as the selected waveform.

SET? (Query Only)

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

**Group**  Miscellaneous

**Related Commands**  HEADer, *LRN?, VERBose

**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 digitizing oscilloscope as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full length.

Examples

a partial return string may look like this:
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;:APPMENU:TITLE "Application Menu";LABEL:BOTTOM1 ";BOTTOM2 ";BOTTOM3 ";BOTTOM4 ";BOTTOM5 ";BOTTOM6 ";BOTTOM7 ";RIGHT1 ";RIGHT2 "; RIGHT3 ";RIGHT4 ";RIGHT5 ";:HEADER 1;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:FORMAT YT;STYLE VECTORS; FILTER SINV;PERSISTENCE 500.0E-3;GRATICULE FULL;TRIGT 1;INTENSITY:OVERALL 85;WAVEFORM 75;TEXT 60;CONTRAST 150;:MESSAGE:SHOW "hello";STATE 1;BOX 74,84,475,135;:LOCK NONE; :HARCOPY:FORMAT EPSIMAGE;PORT GPIB;LAYOUT PORTRAIT;

*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

Related Commands


Syntax

*SRE <NR1>

*SR?

Arguments

<NR1> is a 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.

**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.

<table>
<thead>
<tr>
<th>Group</th>
<th>Status and Error</th>
</tr>
</thead>
</table>

**Related Commands**

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

**Syntax**

*STB?

![Diagram](image)

**Returns**

<NR1>

**Examples**

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

**TEKSecure**

Initializes both waveform and setup memories. This overwrites any previously stored data.

TEKSecure writes zeros in all waveform reference memory, regardless of selected record length, and puts all setups in the factory init state.

TEKSecure then verifies that the waveform and setup memory are in the desired state. It asserts a pass or a fail event on completion.

| Group          | Miscellaneous   |
TIMe

Sets or queries the time that the digitizing oscilloscope can display.

Syntax

TIMe <QString>

Examples

TIME "01:24:00"

specifies that the time is set to 01:24 AM.

TRIGger

Forces a trigger event to occur, and the TRIGger query 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 pressing the front-panel **FORCE TRIGGER** button.

**Examples**

TRIGGER FORCE
forces a trigger event to occur.

TRIGGER?
might return :TRIGGER:MAIN:MODE AUTO;TYPE EDGE;LEVEL -480.0E-3;HOLDOFF:VALUE 0;:TRIGGER:MAIN:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;:TRIGGER:MAIN:LOGIC:CLASS PATTERN;FUNCTION AND;WHEN TRUE;THRESHOLD:CH1 1.40E+0;CH2 1.200E+0;CH3 1.200E+0;CH4 1.200E+0;:TRIGGER:MAIN:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X;:TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4 X;:TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 RISE;:TRIGGER:MAIN:PULSE:CLASS GLITCH;SOURCE CH1;GLITCH:WIDTH 2.0E-9;FILTER ACCEPT;POLARITY POSITIVE;:TRIGGER:MAIN:PULSE:RUNT:POLARITY POSITIVE;THRESHOLD:HIGH 2.00E+0;LOW 800.0E-3;:TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT 2.0E-9:HIGHLIMIT 2.0E-9;WHEN WITHIN;POLARITY POSITIVE;:TRIGGER:DELAY:TYPE EDGE;LEVEL -480.0E-3;BY TIME;EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC;:TRIGGER:DELAY:TIME 16.0E-9;EVENTS:COUNT 2

**TRIGger:DELay**

Sets the delayed trigger level and returns the current delayed trigger parameters.

**Group**

Trigger

**Syntax**

TRIGger:DELay SETLevel

TRIGger:DELay?
Arguments  

SETLevel sets the delayed trigger level to half way between the MIN and MAX amplitudes of the trigger source input. This is equivalent to selecting Set to 50% in the Delayed Edge Level side menu.

Examples  

TRIGGER:DELAY SETLEVEL
sets the delayed trigger level to 50% of MAX and MIN.

TRIGGER:DELAY?
might return:TRIGGER:DELAY:TYPE EDGE;LEVEL 0.0E+0;BY TIME;EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC;;TRIGGER:DELAY:TIME 16.0E-9;EVENTS:COUNT 2

TRIGger:DELaY:BY  

Selects whether the delayed trigger occurs after a specified number of events or a specified period of time after the main trigger. This is equivalent to setting Delay by in the Delayed Trig menu.

Group  

Trigger

Related Commands  

TRIGger:DELaY:EVENTS:COUNt, TRIGger:DELaY:TIme

Syntax  

TRIGger:DELaY:BY
{ EVENTS | TIme | EVENTSTime (TDS 510A, 500B, 600B, & 700A only) | RUNSAfter }

TRIGger:DELaY:BY?
sets the delayed trigger to occur after a set number of trigger events after the main trigger. The number of events is specified by TRIGGER:DELAY:EVENTS:COUNt.

TIME sets the delayed trigger to be ready to occur a set time after the main trigger event. The time period is specified by TRIGGER:DELAY:TIME.

EVENTSTime (TDS 510A, 500B, 600B, & 700A only) sets a specified time after a specified number of delay trigger trigger events — after the main trigger event. For example in examining a pulse train, you might use the main trigger to detect the start of the train, then use the delay by events to go to the position of interest within the pulse train, and then use the time delay to wait a specified time period before starting the data acquisition.

RUNSAfter looks for a main trigger, then waits a user-specified time, then starts acquiring data.

**Examples**
TRIGGER:DELAY:BY?
might return EVENTS.

**TRIGGER:DELAY:EDGE?** (Query Only)

Returns the coupling, slope, and source for the delayed trigger.

**Group** Trigger

**Syntax** TRIGGER:DELAY:EDGE?

Examples
TRIGGER:DELAY:EDGE?
might return :TRIGGER:EDGE:SOURCE CH1;SLOPE RISE;
COUPLING DC

**TRIGGER:DELAY:EDGE:COUPLING**

Selects the type of coupling for the delayed trigger. This command is equivalent to selecting **Coupling** in the Delayed Trig menu.

**Group** Trigger
Syntax  
TRIGger:DELay:EDGE:COUpliNG \{ AC (TDS 400A, 510A) | DC | HFRej (TDS 400A, 510A) | LFRej (TDS 400A, 510A) | MAINTrigg (TDS 500B, 600B & 700A only) | NOISErej \} 

TRIGger:DELay:EDGE:COUpliNG?

Arguments  
AC selects AC trigger coupling (TDS 400A, 510A).
DC selects DC trigger coupling.
HFRej coupling removes the high frequency components of the DC signal (TDS 400A, 510A).
LFRej coupling removes the low frequency components of the AC signal (TDS 400A, 510A).
MAINTrigg coupling sets the delayed trigger coupling to match the setting on the main trigger (TDS 500B, 600B & 700A only).
NOISErej selects DC low sensitivity.

Examples  
TRIGGER:DELAY:EDGE:COUplING DC 
sets the delay trigger to DC coupling.

TRIGGER:DELAY:EDGE:COUplING?
might return LFREJ for the delayed trigger coupling.

TRIGger:DELay:EDGE:SLOpe

Selects either a rising or falling edge for the delayed trigger. This command is equivalent to selecting Slope in the Delayed Trig menu.

Group  
Trigger
**Syntax**

```plaintext
TRIGGER:DELAY:EDGE:SLOpe { RISE | FALL }
```

```plaintext
TRIGGER:DELAY: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:DELAY:EDGE:SLOpe?

might return RISE, indicating that the delayed trigger occurs on the rising edge.

**TRIGGER:DELAY:EDGE:SOURce**

Selects the source for the delayed trigger. This command is equivalent to selecting **Source** in the Delayed Trig menu.

**Group**

Trigger

**Syntax**

```plaintext
TRIGGER:DELAY:EDGE:SOURce { AUXiliary (not available on TDS 520B, 724A, & 782A) | CH<x> }
```

```plaintext
TRIGGER:DELAY:EDGE:SOURce?
```

**Arguments**

AUXiliary specifies an external trigger using the Auxiliary Trigger Input connector that is located on the rear panel of the instrument. The TDS 520B, 724A, & 782A do not have an Auxiliary Trigger input and so do not support this argument.
CH<x> specifies one of the input channels.

**Examples**

TRIGGER:DELAY:EDGE:SOURCE CH1
selects channel 1 as the input source for the delayed trigger.

**TRIGger:DELay:EVENTS? (Query Only)**

Returns the current delayed trigger event parameter.

**Group**

Trigger

**Syntax**

TRIGger:DELay:EVENTS?

**Examples**

TRIGGER:DELAY:EVENTS?
might return :TRIGGER:DELAY:EVENTS:COUNT 2

**TRIGger:DELay:EVENTS:COUNt**

Sets or queries the number of events that must occur before the delayed trigger occurs when TRIGger:DELAY:BY is set to EVENTS. This is equivalent to setting the **Delay by Events** count in the Delayed Edge Delay side menu.

**Group**

Trigger

**Syntax**

TRIGger:DELay:EVENTS:COUNt <NR1>

TRIGger:DELay:EVENTS:COUNt?

**Arguments**

<NR1> is the number of delayed edge trigger events. The TDS 400A range is 1 to (10E7 – 1). The TDS 510A, 500B, 600B, and 700A range is 2 to 10E7.
**Examples**

TRIGGER:DELAY:EVENTS:COUNT 4
specifies that the delayed trigger will occur four trigger events after the main trigger.

TRIGGER:DELAY:EVENTS:COUNT?
might return 2, indicating that two events must occur after the main trigger before the delayed trigger can occur.

---

**TRIGger:DELay:LEVel**

Selects the level of the delayed trigger. This command is equivalent to setting **LEVel** in the Delayed Trig menu.

**Group**
Trigger

**Syntax**
TRIGger:DELay:LEVel { ECL | TTL | <NR3> }

TRIGger:DELay:LEVel?

**Arguments**
ECL specifies a preset ECL level of –1.3 V.

TTL specifies a preset TTL level of 1.4 V.

<NR3> is the delayed trigger level, in volts.

**Examples**
TRIGGER:DELAY:LEVEL 2E-3
sets the delayed trigger level to 2 mV.

---

**TRIGger:DELay:TIMe**

Sets or queries the delay time when HORizontal:DELay:MODE is set to TRIGAfter. This command is identical to the HORizontal:DELay:TIME:TRIGAfter command, and is equivalent to setting the Delay by Time value in the Delayed Edge Delay side menu.

When HORizontal:DELay:MODE is set to RUNSAfter, the delay time is set by the HORizontal:DELay:TIME:RUNSAfter command.
**TRIGger:DELay:TIME**

Sets or queries the delay time.

**Group** Trigger

**Related Commands** VHORizontal:DELay:MODe, VHORizontal:DELay:TIME:RUNSAfter, VHORizontal:DELay:TIME:TRIGAfter

**Syntax**

TRIGger:DELay:TIME <NR3>

TRIGger:DELay:TIME?

**Arguments**

<NR3> is the delay time, in seconds.

**Examples**

TRIGGER:DELAY:TIME 4E-6

sets the delay time to 4 μs.

**TRIGger:DELay:TYPE**

Sets or queries the type of delayed trigger.

**Group** Trigger

**Syntax**

TRIGger:DELay:TYPE EDGE

TRIGger:DELay:TYPE?

**Arguments**

EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction. Use the TRIGger:DElay:LEVel and TRIGger:DElay:EDGE:SLOpe commands to set the voltage level and direction respectively.

**Examples**

TRIGGER:DELAY:TYPE?

always returns EDGE as the type of delayed trigger.
**TRIGger:MAIn**

Sets the main trigger level and returns the current main trigger parameters.

**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.

**Examples**  
TRIGGER:MAIN SETLEVEL
sets the main trigger level mid way between MAX and MIN.

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

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

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:EDGE?

**Examples**  
TRIGGER:MAIN:EDGE?
might return SOURCE CH1;COUPLING DC;SLOPE RISE

**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 trigger 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.

**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?
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**

```
TRIGger:MAIN:EDGE:SOURce { AUXiliary (not available on TDS 520B, TDS 724A, & TDS 782A) | CH<x> | LINE }
```

```
TRIGger:MAIN:EDGE:SOURce?
```

**Arguments**

- AUXiliary specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the oscilloscope. The TDS 520B, 724A, and 782A do not have an Auxiliary Trigger input and so do not support this argument.
- CH<x> specifies one of the input channels.
- LINE specifies AC line voltage.

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.
Examples  TRIGGER:MAIN:EDGE:SOURCE LINE
specifies the AC line voltage as the main edge trigger source.

TRIGGER:MAIN:EDGE:SOURCE?
might return CH2 for the main edge trigger source.

TRIGger:MAIn:HOLDOff? (Query Only)
For TDS 500B, 600B and 700A, returns the main trigger holdoff default (TIME or DEFAULT) and main trigger holdoff time. For the TDS 400A and 510A, returns the main trigger holdoff value.

Group  Trigger

Syntax  TRIGger:MAIn:HOLDOff?

Examples  TRIGGER:MAIN:HOLDOFF?
for the TDS 500B, 600B and 700A, might return
:TRIGGER:MAIN:HOLDOFF:TIME 250.0E-9;BY DEFAULT

or, for TDS 400A and 510A, might return
:TRIGGER:MAIN:HOLDOFF:VALUE 0.

TRIGger:MAIn:HOLDOff:ACTUal? (Query Only)
* TDS 500B, 600B and 700A Only *

Returns the main trigger holdoff value in seconds. This is equivalent to selecting Mode & Holdoff from the main Trigger menu and viewing the value in the side menu Holdoff Default or Holdoff Time items (whichever is highlighted).

Group  Trigger

Syntax  TRIGger:MAIn:HOLDOff:ACTUal?
Examples  TRIGGER:MAIN:HOLDOFF:ACTUAL?
might return 4E–6 showing the holdoff time is set to 4 μs.

TRIGGER:MAIN:HOLDOFF:BY
*TDS 500B, 600B & 700A Only*

Sets or queries the main trigger holdoff default. This is equivalent to selecting Mode & Holdoff from the main Trigger menu, then setting Default Holdoff or Holdoff (Time) in the resulting side menu.

Group  Trigger

Syntax  TRIGGER:MAIN:HOLDOFF:BY { TIME | DEFAult }

Examples  TRIGGER:MAIN:HOLDOFF:BY TIME
sets the holdoff to the by time setting. This enables the user to set the holdoff time.

TRIGGER:MAIN:HOLDOFF:TIME
*TDS 500B, 600B & 700A Only*

Sets or queries the main trigger holdoff time. This is equivalent to setting Holdoff Time in the Mode & Holdoff side menu.

Group  Trigger
Syntax

TRIGGER:MAIN:HOLDOFF:TIME <NR3>
TRIGGER:MAIN:HOLDOFF:TIME?

Arguments

<NR3> holdoff time in seconds. The range is 250 ns to 12.0 seconds.

Examples

TRIGGER:MAIN:HOLDOFF:TIME 10
sets the holdoff time to be 10 seconds.

TRIGGER:MAIN:HOLDOFF:VALUe

Sets or queries the main trigger holdoff value. This is equivalent to setting Holdoff in the Mode & Holdoff side menu.

Group

Trigger

Syntax

TRIGGER:MAIN:HOLDOFF:VALUe <NR1>
TRIGGER:MAIN:HOLDOFF:VALUe?

Arguments

<NR1> is from 0 to 100, and is a percent of the holdoff range.

Examples

TRIGGER:MAIN:HOLDOFF:VALUE 10
set the holdoff value to be 10% of the holdoff range.

TRIGGER:MAIN:LEVEL

Sets the main trigger level. This command is equivalent to adjusting the front-panel TRIGGER MAIN LEVEL knob.
**Command Descriptions**

**Group** Trigger

**Syntax**

```
TRIGger:MAIn:LEVel { ECL | TTL | <NR3> }
```

```
TRIGger:MAIn:LEVel?
```

**Arguments**

- **ECL** specifies a preset ECL level of –1.3 V.
- **TTL** specifies a preset TTL level of 1.4 V.
- **<NR3>** is the main trigger level, in volts.

**Examples**

```
TRIGGER:MAIN:LEVEL?
```

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

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

*TD5 510A, 500B, 600B, & 700A Only*

Returns all main logic trigger parameters.

**Group** Trigger

**Syntax**

```
TRIGger:MAIn:LOGIc?
```

**Examples**

```
TRIGGER:MAIN:LOGIC?
```

might return:

```
:TRIGGER:MAIN:LOGIC:CLASS: PATTERN;
FUNCTION: AND; WHEN: TRUE; THRESHOLD: CH1 0; CH2 0; CH3 0; CH4 0;
:TRIGGER:MAIN:LOGIC:INPUT: CH1: HIGH; CH2: X; CH3: X;
```
TRIGger:MAIn:LOGIc:CLAss

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the type of main logic trigger. This command is equivalent to selecting **Class** in the Trigger menu when the **Type** is set to Logic.

**Group**
Trigger

**Syntax**
TRIGger:MAIn:LOGIc:CLAss { PATtern | STATE
SETHold (TDS 500B, 600B & 700A only)}

TRIGger:MAIn:LOGIc:CLAss?

**Arguments**
PATtern means that the instrument triggers when the specified logical combinations of channels 1, 2, 3, and 4 are met.

STATE means that the instrument triggers when the specified conditions of channels 1, 2, and 3 are met after the channel 4 condition is met.

SETHold means the oscilloscope will trigger on the setup and hold violations between a data source and a clock source (TDS 500B, 600B & 700A only).

**Examples**
TRIGGER:MAIN:LOGIC:CLASS?
might return STATE.

TRIGger:MAIn:LOGIc:FUNCTION

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the logical combination of the input channels for the main logic trigger.

When TRIGger:MAIn:LOGIc:CLAss is PATtern, this command applies to channels 1, 2, 3, and 4. When TRIGger:MAIn:LOGIc:CLAss is STATE, only channels 1, 2, and 3 are logically combined. This command is equivalent to selecting the function in the Logic Pattern Function side menu.
Group  Trigger

Syntax  TRIGger:MAIn:LOGIc:FUNCTION { AND | NAND | NOR | OR }
       TRIGger:MAIn:LOGIc:FUNCTION?

Arguments  AND specifies that the instrument will trigger if all the conditions are true.
NAND specifies that the instrument will trigger if any of the conditions are false.
NOR specifies that the instrument will trigger if all of the conditions are false.
OR specifies that the instrument will trigger if any of the conditions are true.

Examples  TRIGGER:MAIN:LOGIC:FUNCTION NOR
sets the logical combination of channels to be true when none of the conditions are true.
TRIGGER:MAIN:LOGIC:FUNCTION?
might return NAND.

TRIGger:MAIn:LOGIc:INPut? (Query Only)
TDS 510A, 500B, 600B, & 700A Only

Returns the main logic trigger input for all channels.

Group  Trigger

Syntax  TRIGger:MAIn:LOGIc:INPut?

Examples  TRIGGER:MAIN:LOGIC:INPUT?
might return :TRIGGER:MAIN:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X
TRIGger:MAIn:LOGIc:INPUT:CH<x>

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the main logic trigger input for the specified channel. The channel is specified by <x> and is 1, 2, or 3. This is equivalent to setting the inputs in the Logic Pattern Inputs side menu.

**Group**
Trigger

**Related Commands**
TRIGger:MAIn:LOGIc:CLAss

**Syntax**
TRIGger:MAIn:LOGIc:INPUT:CH<x> \{ HIGH | LOW | X \}
TRIGger:MAIn:LOGIc:INPUT:CH<x>?

**Arguments**
HIGH specifies logic high.
LOW specifies logic low.
X specifies a do not care state.

**Examples**
TRIGGER:MAIN:LOGIC:INPUT:CH2 LOW
sets the main logic trigger input to logic low for channel 2.

TRIGger:MAIn:LOGIc:PATtern:INPUT:CH4

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the main logic trigger input for channel 4. These are the inputs used when TRIGger:MAIn:LOGIc:CLAss is set to PATtern. This is equivalent to setting the channel 4 input in the Logic Pattern Inputs side menu.

**Group**
Trigger

**Related Commands**
TRIGger:MAIn:LOGIc:CLAss
**Syntax**

TRIGger:MAIn:LOGIc:PAIttern:INPut:CH4 \{ HIGH | LOW | X \}

TRIGger:MAIn:LOGIc:PAIttern:INPut:CH4?

**Arguments**

HIGH specifies logic high.

LOW specifies logic low.

X specifies a do not care state.

**Examples**

TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4 LOW

sets the main logic trigger input to logic low for channel 4 when the logic class is set to PAttern.

TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4?

might return X, indicating that the logic input for channel 4 is do not care.

**TRIGger:MAIn:LOGIc:PAIttern:WHEn**

*TDS 510A, 500B 600B, & 700A Only*

Sets or queries a condition for generating a main logic pattern trigger.

**Group**

Trigger

**Syntax**

TRIGger:MAIn:LOGIc:PAIttern:WHEn \{ TRUE | FALSE | LESSThan | MOREThan \}

TRIGger:MAIn:LOGIc:PAIttern:WHEn?
**Arguments**

TRUE specifies the trigger to occur when the pattern becomes true.

FALSE specifies the trigger to occur when the pattern becomes false.

LESSThan specifies trigger to occur if the specific pattern is true less than the LESSLimit. (see Figure 2–5 and TRIGger:MAIn:LOGic:PATtern:WHEn:LESSLimit) Trigger is evaluated at the true-false transition.

MOREThan specifies trigger to occur if the specific pattern is true longer than the more limit. (see Figure 2–5 and TRIGger:MAIn:LOGic:PATtern:WHEn:MORELimit) Trigger is evaluated at the true-false transition.

![Figure 2–5: LESSThan and MOREThan Arguments](image_url)

**TRIGger:MAIn:LOGic:PATtern:WHEn:LESSLimit**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the maximum time the selected pattern may be true and still generate a main logic pattern trigger.

**Group** Trigger

**Syntax**

```
TRIGger:MAIn:LOGic:PATtern:WHEn:LESSLimit <NR3>
TRIGger:MAIn: LOGic: PATtern: WHEN: LESSLimit?
```
TRIGger:MAIn:LOGIc:PAttern:WHen:MORELimit

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the minimum time the selected pattern may be true and still generate a main logic pattern trigger.

**Group**
Trigger

**Syntax**

```
TRIGger:MAIn:LOGIc:PAttern:WHen:MORELimit <NR3>
TRIGger:MAIn:LOGIc:PAttern:WHen:MORELimit?
```

**Arguments**

<NR3> time to hold pattern true.

TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE

*TDS 500B, 600B & 700A only*

Sets or queries the clock edge polarity for setup and hold violation triggering. This is equivalent to selecting Define Clock from the main Trigger menu and Polarity in the resulting side menu.

**Group**
Trigger

**Syntax**

```
TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE { FALL | RISE }
```

**Arguments**

<NR3> time to hold pattern true.
TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE?

Arguments
FALL specifies falling edge.
RISe specifies rising edge.

Examples
TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:EDGE RISE specifies the polarity as the rising edge.

TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel
*TDS 500B, 600B & 700A only*

Sets or queries the main logic setup/hold clock voltage trigger level. This is equivalent to selecting Levels from the main Trigger menu and Clock Level in the resulting side menu.

Group
Trigger

Syntax
TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel { ECL | TTL | <NR3> }
TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel?

Arguments
ECL specifies a preset ECL level of −1.3 V.
TTL specifies a preset TTL level of 1.4 V.
<NR3> is the main trigger level, in volts.
**Examples**

TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:LEVEL 1.4
sets the main logic trigger setup/hold clock level to 1.4 volts.

---

**TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:SOURce**

*TDS 500B, 600B & 700A Only*

Sets or queries the source for the clock for the main logic trigger setup/hold input. The channel is specified by <x> and is 1, 2, 3, or 4. This is equivalent to selecting **Define Clock** from the main Trigger menu and **CH1**, **CH2**, **CH3**, or **CH4** in the resulting side menu.

**Group**

Trigger

**Related Commands**

TRIGGER:MAIN:LOGIC:CLASS

**Syntax**

TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:SOURce: CH<x>

TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:SOURce?

**Arguments**

CH<x> specifies one of the input channels (CH1, CH2, CH3, or CH4).

**Examples**

TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:SOURce CH2
selects Channel 2 as the clock source for the main logic trigger setup/hold.

---

**TRIGGER:MAIN:LOGIC:SETHOLD:DATA:LEVEL**

*TDS 500B, 600B & 700A only*

Sets or queries the main logic set/hold data level. This is equivalent to selecting **Levels** from the main Trigger menu and **Data Level** in the resulting side menu.

**Group**

Trigger

**Syntax**

TRIGGER:MAIN:LOGIC:SETHOLD:DATA:LEVEL \{ ECL | TTL | <NR3> \}
**TRIGger:MAIn:LOGIc:SETHold:DATa:LEVEL?**

- **Arguments**
  - ECL specifies a preset ECL level of –1.3 V.
  - TTL specifies a preset TTL level of 1.4 V.
  - <NR3> is the main trigger level, in volts.

- **Examples**
  - `TRIGGER:MAIN:LOGIC:SETHold:DATA:LEVEL 1.4`
    sets the main logic setup/hold data level to 1.4 volts.

**TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce**

*TDS 500B, 600B & 700A Only*

Sets or queries the data channel for the main logic trigger set/hold input. The channel is specified by `<x>` and is 1, 2, 3, or 4. This is equivalent to selecting **Data Source** from the main Trigger menu and **CH1**, **CH2**, **CH3**, or **CH4** in the resulting side menu.

- **Group**
  - Trigger

- **Related Commands**
  - TRIGger:MAIn:LOGIc:CLAss

- **Syntax**
  - `TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce CH<x>`
  - `TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce?`

- **Arguments**
  - `CH<x>` specifies one of the input channels (CH1, CH2, CH3, or CH4)
**Command Descriptions**

**Examples**

TRIGGER:MAIN:LOGIC:SETHOLD:DATA:SOURCe CH2
selects Channel 2 as the source for the main logic trigger set/hold.

**TRIGger:MAIn:LOGIc:SETHold:HOLDTime**

*TDS 500B, 600B & 700A only*

Sets or queries the main logic trigger hold time. This is equivalent to selecting **Setup/Hold Times** from the main Trigger menu and **Hold Time** in the resulting side menu.

**Group**

Trigger

**Syntax**

TRIGGER:MAIn:LOGIC:SETHold:HOLDTime <NR3>

TRIGGER:MAIn:LOGIC:SETHold:HOLDTime?

**Arguments**

<NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

**Examples**

TRIGGER:MAIn:LOGIC:SETHold:HOLDTIME 200 E-12
sets the main logic trigger set/hold holdtime to 200 nanoseconds.

**TRIGger:MAIn:LOGIc:SETHold:SETTTime**

*TDS 500B, 600B & 700A only*

Sets or queries the main logic trigger set time. This is equivalent to selecting **Setup/Hold Times** from the main Trigger menu and **Setup Time** in the resulting side menu.

**Group**

Trigger

**Syntax**

TRIGGER:MAIn:LOGIC:SETHold:SETTTime <NR3>

TRIGGER:MAIn:LOGIC:SETHold:SETTTime?
Arguments  

<NR3> specifies the setup time setting in seconds. Positive values occur before the clock edge. Negative values occur after the clock edge.

Examples  

TRIGGER:MAIN:LOGIC:SETTHOLD:SETTIME 600 E-12 sets the main logic trigger sethold time to 600 nanoseconds.

**TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4**  
*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the main logic trigger input for channel 4. This input is used when TRIGGER:MAIN:LOGIC:CLASS is set to STATE. This is equivalent to setting the channel 4 input in the Logic Pattern Inputs side menu.

**Group**  
Trigger

**Syntax**  
TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 { FALL | RISE }
TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4?

Arguments  

FALL specifies falling edge.
RISe specifies rising edge.

Examples  

TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 RISE specifies that the main logic trigger input for channel 4 is the rising edge when the logic class is set to STATE.
TRIGger:MAIn:LOGIc:STATE:WHEn
*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the main logic state trigger.

**Group**  Trigger

**Syntax**  TRIGger:MAIn:LOGIc:STATE:WHEn { TRUE | FALSE }

TRIGger:MAIn:LOGIc:STATE:WHEn?

**Arguments**  
TRUE specifies the trigger to occur when the condition is met on the fourth channel and the pattern of the first three channels are at the desired states.

FALSE

TRIGger:MAIn:LOGIc:THReshold? (Query Only)
*TDS 510A, 500B, 600B, & 700A Only*

Returns the main logic trigger threshold voltage for all channels.

**Group**  Trigger

**Syntax**  TRIGger:MAIn:LOGIc:THReshold?

**Examples**  
TRIGGER:MAIN:LOGIC:THRESHOLD? might return :TRIGGER:MAIN:LOGIC:THRESHOLD:CH1 0;CH2 0; CH3 0;CH4 0
**TRIGger:MAIn:LOGic:THReshold:CH<x>**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the main logic trigger threshold voltage for the channel specified by <x>. This is equivalent to setting the thresholds in the Logic State Threshold and Logic Pattern Threshold side menus.

**Group**  
Trigger

**Syntax**

TRIGger:MAIn:LOGic:THReshold:CH<x> <NR3>

TRIGger:MAIn:LOGic:THReshold:CH<x>?

**Arguments**

<NR3> specifies the threshold voltage.

**Examples**

TRIGGER:MAIN:LOGIC:THRESHOLD:CH1 .5  
sets the main logic trigger threshold for channel 1 to .5 volts.

**TRIGger:MAIn:LOGic:WHEn**

*TDS 510A, 500B, 600B, & 700A Only*

Specifies whether the main logic trigger occurs when the specified state goes true or false when TRIGger:MAIn:LOGic:CLAss is set to PATtern. This is equivalent to selecting **Trigger When** in the Trigger menu.

**Group**  
Trigger

**Syntax**

TRIGger:MAIn:LOGic:WHEn { FALSE | TRUE }

TRIGger:MAIn:LOGic:WHEn?
Examples  TRIGGER:MAIN:LOGIC:WHEN TRUE
specifies that the main logic trigger when the logic pattern is true.

TRIGGER:MAIN:MODE

Sets or queries the main trigger mode. This command is equivalent to selecting Mode & Holdoff in the Trigger menu.

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.
NORMAL waits for a valid trigger event.

Examples  TRIGGER:MAIN:MODE AUTO
specifies that a trigger event is automatically generated.

TRIGGER:MAIN:PULSe? (Query Only)

*TDS 510A, 500B, 600B, & 700A Only*

Returns the main pulse trigger parameters.

Group  Trigger
**TRIGGER:MAIN:PULSE?**
might return `TRIGGER:MAIN:PULSE:CLASS GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0E-9;FILTER ACCEPT;POLARITY POSITIVE;:TRIGGER: MAIN:PULSE:RUNT:POLARITY POSITIVE;THRESHOLD:HIGH 2.00E+0;LOW 800.0E-3;:TRIGGER:MAIN:PULSE:LOWLIMIT 2.0E-9;HIGHLIMIT 2.0E-9;WHEN WITHIN;POLARITY POSITIVE` as the current main pulse trigger parameters.

**TRIGGER:MAIN:PULSE:CLASS**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the type of pulse to trigger on. This command is equivalent to selecting **Class** in the Trigger menu.

**Group**
Trigger

**Syntax**
```plaintext
TRIGger:MAIn:PULse:CLAss { GLItch | RUNT | WIDth | SLEWRate (TDS 500B, 600B & 700A only) | TIMEOUT (TDS 500B, 600B, 700A only) }
TRIGger:MAIn:PULse:CLAss?
```

**Arguments**
- **GLItch** triggers when a pulse is found that is of the specified polarity and width. These are set with the commands `TRIGger:MAIn:PULse:GLItch:POLarity` and `TRIGger:MAIn:PULse:GLItch:WIDth`.
- **RUNT** triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the `TRIGger:MAIn:PULse:RUNT:THReshold:LOW` and `TRIG-`
ger:MAIn:PULse:RUNT:THReshold:HIGH commands. The crossing can be either positive or negative as specified by TRIGger:MAIn:PULse: RUNT:PO-Larity.

WIDTH triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by TRIGger:MAIn:PULse: WIDth:LOW-Limit and TRIGger:MAIn:PULse:WIDth:HIGHLimit. The polarity is selected using the TRIGger:MAIn:PULse:WIDth:POLarity command.

SLEWRate triggers when the slew rate of the source violates the specified conditions (TDS 500B, 600B & 700A only).

TIME0ut triggers when the pulse train stops in the selected state for longer than the specified time (TDS 500B, 600B, & 700A only).

**Examples**

```
TRIGGER:MAIN:PULSE:CLASS WIDTH
```
specifies a width pulse for the main trigger.

### TRIGger:MAIn:PULse:GLItch? (Query Only)

**TDS 510A, 500B, 600B, & 700A Only**

Returns the current main glitch pulse trigger parameters.

**Group** Trigger

**Syntax** TRIGger:MAIn:PULse:GLItch?

```
TRIGGER:MAIN:PULse:GLItch?
```

**Examples**

```
might return: TRIGGER:MAIN:PULSE:CLASS GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0E-9;FILTER ACCEPT;POLARITY POSITIVE.
```

### TRIGger:MAIn:PULse:GLItch:FILTer

**TDS 510A, 500B, 600B, & 700A Only**

Controls glitch detection. This command is equivalent to selecting Filter in the Trigger menu.

**Group** Trigger
specifies that the digitizing oscilloscope will trigger only on pulses that are narrower than the specified width when the main trigger type is set to pulse glitch. The width is specified using TRIGger:MAIn:PULse:GLItch:WIDth command.

REJect specifies that the digitizing oscilloscope will trigger only on pulses that are wider than the specified width when the main trigger type is set to pulse glitch. The width is specified using TRIGger:MAIn:PULse:GLItch:WIDth command.

returns either Accept or REject, indicating whether glitches are filtered.

TRIGger:MAIn:PULse:GLItch:POLarity

Sets or queries the polarity for the main pulse glitch trigger. This command is equivalent to selecting Polarity & Width in the Trigger menu.

Group Trigger

Syntax TRIGger:MAIn:PULse:GLItch:POLarity { EITHER | NEGAtive | POSITIVE }

TRIGger:MAIn:PULse:GLItch:POLarity?
**Examples**

TRIGGER:MAIN:PULSE:GLITCH:POLARITY EITHER specifies that the polarity of the glitch can be either positive or negative.

**TRIGger:MAIn:PULse:GLItch:WIDth**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the width for the main pulse glitch trigger. This command is equivalent to selecting **Polarity & Width** in the Trigger menu.

**Group**

Trigger

**Syntax**

TRIGger:MAIn:PULse:GLItch:WIDth <NR3>

TRIGger:MAIn:PULse:GLItch:WIDth?

**Arguments**

<NR3> is the width of the glitch, in seconds.

**Examples**

TRIGGER:MAIN:PULSE:GLITCH:WIDTH 15E-6 sets the width of the glitch to 15 µs.

**TRIGger:MAIn:PULse:RUNT? (Query Only)**

*TDS 510A, 500B, 600B, & 700A Only*

Returns the current parameters for the main pulse runt trigger.
**TRIGger:MAIn:PULse:RUNT:POLarity**  
*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the polarity for the main pulse runt trigger. This command is equivalent to selecting **Polarity** in the Trigger menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:RUNT:POLarity { EITHer | NEGative | POSITIve }

TRIGger:MAIn:PULse:RUNT:POLarity?

**Arguments**  
NEGative indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.

POSITIve indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

EITHer indicates either NEGative or POSITIve polarity.

**Examples**  
TRIGGER:MAIn:PULSE:RUNT:POLARITY NEGATIVE specifies that the polarity of the main pulse runt trigger is negative.
TRIGger:MAIN:PULSE:RUNT:THReshold? (Query Only)
TDS 510A, 500B, 600B, & 700A Only

Returns the upper and lower thresholds for the main pulse runt trigger.

Group Trigger

Syntax TRIGger:MAIN:PULSE:RUNT:THReshold?

Examples TRIGGER:MAIN:PULSE:RUNT:THRESHOLD?
might return :TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:HIGH 2.00E+0;LOW 800.0E-3.

TRIGger:MAIN:PULSE:RUNT:THReshold:BOTh
TDS 500B, 600B, & 700A Only

Sets or queries the trigger level switching thresholds for the main pulse runt trigger. This command is equivalent to setting Set to TTL or Set to ECL in the Main Pulse Runt Trigger menu’s Thresholds side menu item.

Group Trigger

Syntax TRIGger:MAIN:PULSE:RUNT:THReshold:BOTh { TTL | ECL }

Arguments TTL sets the upper threshold to 1.8 V and the lower threshold to 0.8 V.
ECL sets the upper threshold to −1.1 V and the lower threshold to −1.5 V.

Examples TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:BOTh TTL
sets the threshold of the pulse runt trigger to the nominal TTL voltage levels.
**TRIGger:MAIn:PULse:RUNT:THReshold:HIGH**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the upper limit for the main pulse runt trigger. This command is equivalent to setting the threshold in the Pulse Runt Threshold side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:RUNT:THReshold:HIGH <NR3>  
TRIGger:MAIn:PULse:RUNT:THReshold:HIGH?

**Arguments**  
<NR3> is the threshold, in volts.

**Examples**  
TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:HIGH 120E-3  
sets the upper limit of the pulse runt trigger to 120 mV.

**TRIGger:MAIn:PULse:RUNT:THReshold:LOW**

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the lower limit for the main pulse runt trigger. This command is equivalent to setting the threshold in the Pulse Runt Threshold side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:RUNT:THReshold:LOW <NR3>  
TRIGger:MAIn:PULse:RUNT:THReshold:LOW?
Arguments  

<NR3> is the threshold, in volts.

Examples  

TRIGGER:MAIn:PULse:RUNT:THRESHold:LOW 50E-3
sets the lower limit of the pulse runt trigger to 50 mV.

**TRIGger:MAIn:PULse:RUNT:WHEn**

*TDS 500B, 600B & 700A Only*

Sets or queries the type of pulse width the trigger checks for when it uncovers a runt. This is equivalent to selecting Trigger When from the main Trigger’s Pulse, Runt menu and Occurs or Wider Than in the resulting side menu.

<table>
<thead>
<tr>
<th>Group</th>
<th>Trigger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>TRIGGER:MAIn:PULse:RUNT:WHEn { OCCurs</td>
</tr>
<tr>
<td></td>
<td>TRIGGER:MAIn:PULse:RUNT:WHEN?</td>
</tr>
</tbody>
</table>

Arguments  

OCCurs specifies a trigger if a runt of any detectable width occurs.
WIDERthan specifies a trigger if a runt of greater than the specified width occurs.

Examples  

TRIGGER:MAIn:PULse:RUNT:WHEN WIDERthan
sets the runt trigger to occur when the oscilloscope detects a runt in a pulse wider than the specified width.
**TRIGger:MAIn:PULse:RUNT:WIDth**  
*TDS 500B, 600B, & 700A Only*

Sets or queries the minimum width for a valid main pulse runt trigger. This command is equivalent to entering a value in the Trigger menu’s `Wider Than` side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:RUNT:WIDth <NR3>
TRIGger:MAIn:PULse:RUNT:WIDth?

**Arguments**  
<NR3> is the minimum width in seconds.

**Examples**  
TRIGGER:MAIN:PULSE:RUNT:WIDTH 15E-6
sets the minimum width of the pulse runt trigger to 15 μs.

**TRIGger:MAIn:PULse:SLEWRate:DELTATime**  
*TDS 500B, 600B, & 700A Only*

Sets or queries the delta time used in calculating the slew rate trigger. This is equivalent to selecting `Trigger When` from the main Trigger’s Slew Rate menu and `Delta Time` in the resulting side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:SLEWRate:DELTATime <NR3>
TRIGger:MAIn:PULse:SLEWRate:DELTATime?
Arguments  <NR3> is the delta time in seconds.

Examples  TRIGGER:MAIN:PULSE:SLEWRATE:DELTATIME 15E-6
sets the slew rate trigger’s delta time to 15 μs.

TRIGGER:MAIN:PULSE:SLEWRATE:POLarity
*TDS 500B, 600B & 700A Only*

Sets or queries the polarity for the main pulse slew rate trigger. This command is equivalent to selecting *Polarity* in the Trigger menu with *Type* set to *SlewRate*.

**Group**  Trigger

**Syntax**  TRIGGER:MAIN:PULSE:SLEWRATE:POLarity { EITHER | NEGAtive | POSITIVE }
TRIGGER:MAIN:PULSE:SLEWRATE:POLarity?

Arguments  NEGAtive indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for slew rate triggering to occur.

POSITIVE indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for slew rate triggering to occur.

EITHER indicates either NEGAtive or POSITIVE polarity.

Examples  TRIGGER:MAIN:PULSE:SLEWRATE:POLARITY EITHER
specifies that the polarity of the slew rate can be either positive or negative.
**TRIGger:MAIn:PULse:SLEWRate:SLEWRate? (Query Only)**

*TDS 500B, 600B, & 700A Only*

Returns the slew rate value. This is the

\[
\frac{(\text{Upper Threshold} - \text{Lower Threshold})}{\text{Delta Time}}
\]

The value is limited to the three most significant digits.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:SLEWRate:SLEWrate?

**Examples**  
TRIGGER:MAIN:PULSE:SLEWRATE:SLEWRATE?

returns the slew rate. It is given as an appropriate amount of volts per second. For example, 1.2E+6 would represent a slew rate of 1.2 megavolts/second.

**TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh**

*TDS 500B, 600B, & 700A Only*

Sets the upper and lower slew rate trigger thresholds. This is equivalent to selecting **Thresholds** from the main Trigger’s Slew Rate menu and entering a value in the resulting side menu’s **High Threshold** or **Low Threshold** items.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh { TTL | ECL }

**Arguments**  
TTL sets the upper threshold to 1.8 V and the lower threshold to 0.8 V.  
ECL sets the upper threshold to –1.1 V and the lower threshold to –1.5 V.
Examples TRIGGER:MAIN:PULSE:SLEWRATE:THRESHOLD:THRESHOLD:HIGH TTL sets the trigger threshold to TTL.

TRIGGER:MAIN:PULSe:SLEWRate:THReshold:HIGh

Sets or queries the upper (most positive) limit of the two threshold levels that a pulse edge must traverse for the slew rate trigger to occur. This command is equivalent to setting the higher threshold in the Pulse Slew Rate Trigger’s Thresholds side menu item.

Group Trigger

Syntax TRIGGER:MAIN:PULSe:SLEWRate:THReshold:HIGh <NR3>

TRIGGER:MAIN:PULSe:SLEWRate:THReshold:HIGh?

Arguments <NR3> is the threshold, in volts.

Examples TRIGGER:MAIN:PULSe:SLEWRate:THReshold:HIGh 120E-3 sets the upper limit of the pulse slew rate trigger to 120 mV.

TRIGGER:MAIN:PULSe:SLEWRate:THReshold:LOW

Sets or queries the lower (most negative) limit of the two threshold levels that a pulse edge must traverse for the slew rate trigger to occur. This command is equivalent to setting the lower threshold in the Pulse Slew Rate Trigger’s Thresholds side menu item.

Group Trigger

Syntax TRIGGER:MAIN:PULSe:SLEWRate:THReshold:LOW <NR3>
TRIGger:MAIn:PULse:SLEWRate:THreshold:LOW?

Arguments

<NR3> is the threshold, in volts.

Examples

TRIGger:MAIN:PULSE:SLEWRate:THRESHold:LOW 50E-3
sets the lower limit of the pulse slew rate trigger to 50 mV.

TRIGger:MAIn:PULse:SLEWRate:WHEn

*TDS 500B, 600B, & 700A Only*

Sets or queries whether to check for a slewing signal that is faster or slower than the specified delta time. This is equivalent to selecting **Trigger When** from the main Trigger’s Slew Rate menu and entering a value in the resulting side menu’s **Trig if Less Than** or **Trig if Greater Than** items.

Group

Trigger

Syntax

TRIGger:MAIN:PULse:SLEWRate:WHEn { FASTERthan | SLOWERthan }

TRIGger:MAIN:PULse:SLEWRate:WHen?

Arguments

FASTERthan sets the trigger to occur when the slew is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the slew is slower than the set volts/second rate.
Examples  TRIGGER:MAIN:PULSE:SLEWRATE:WHEN FASTERTHAN sets the slew rate trigger to work when the slew is faster than the set volts/second rate.

**TRIGger:MAIn:PULse:SOUrce**  
*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the source for the main pulse trigger. This is equivalent to selecting the source in the Pulse Runt Source side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:SOUrce CH<x>  
TRIGger:MAIn:PULse:SOUrce?

**Examples**  
TRIGGER:MAIN:PULSE:SOURCE CH2 selects Channel 2 as the source for the main pulse trigger.

**TRIGger:MAIn:PULse:TIMEOut:POLarity**  
*TDS 500B, 600B, & 700A Only*

Sets or queries the polarity for the main pulse timeout trigger. This command is equivalent to selecting Polarity in the Trigger menu with Type set to Time out.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:TIMEOut:POLarity { STAYSHigh (or POSITIVe) | STAYSLow (or NEGATive) | EITHER }
TRIGger:MAIn:PULse:TIMEOut:POLarity?
Arguments

STAYSLow (or NEGative) indicates that a pulse edge must stay low the required time period for timeout triggering to occur.

STAYSHigh (or POSITIVE) indicates that a pulse edge must stay high the required time period for timeout triggering to occur.

EITHER indicates either STAYSHigh or STAYSLow polarity.

Examples

TRIGGER:MAIN:PULSE:TIMEOUT:POLARITY EITHER specifies that the polarity of the timeout trigger can be either positive or negative.

TRIGger:MAIn:PULse:TIMEOut:TIMe

*TDS 500B, 600B, & 700A Only*

Sets or queries the pulse timeout trigger time, in seconds. This command is equivalent to selecting Time in the Trigger menu and entering a value with the keypad or general purpose knob.

Group

Trigger

Syntax

TRIGger:MAIn:PULse:TIMEOut:TIMe <NR3>

TRIGger:MAIn:PULse:TIMEOut:TIME?

Arguments

<NR3> is the timeout time period, in seconds.
Examples

sets the timeout time to 3.134 μs.

TRIGGER:MAIN:PULSE:WIDth? (Query Only)
TDS 510A, 500B, 600B, & 700A Only

Returns the width parameters for the main pulse width trigger.

Group
Trigger

Syntax
TRIGGER:MAIN:PULSE:WIDth?

Examples

TRIGGER:MAIN:PULSE:WIDth?
might return TRIGGER:MAIN:PULSE:WIDth:LOWLIMIT 2.0E-9; HIGHLIMIT 2.0E-9; WHEN WITHIN; POLARITY POSITIVE as the current main pulse trigger parameters.

TRIGGER:MAIN:PULSE:WIDth:HIGHLimit
TDS 510A, 500B, 600B, & 700A Only

Sets or queries the upper limit for the main pulse width trigger. This is equivalent to setting Upper Limit in the Pulse Width trig when side menu.

Group
Trigger

Syntax
TRIGGER:MAIN:PULSE:WIDth:HIGHLimit <NR3>
TRIGGER:MAIN:PULSE:WIDth:HIGHLimit?

Arguments
<NR3> is the upper limit, in seconds.
TRIGger:MAIn:PULse:WIDth:LOWLimit

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the lower limit for the main pulse width trigger. This is equivalent to setting **Lower Limit** in the Pulse Width Trig When side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:WIDth:LOWLimit <NR3>
TRIGger:MAIn:PULse:WIDth:LOWLimit?

**Arguments**  
<NR3> is the lower limit, in seconds.

TRIGger:MAIn:PULse:WIDth:POLarity

*TDS 510A, 500B, 600B, & 700A Only*

Sets or queries the polarity for the main pulse width trigger. This is equivalent to selecting the polarity in the Pulse Width Polarity side menu.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:PULse:WIDth:POLarity { NEGAtive | POSITIve }
TRIGger:MAIn:PULse:WIDth:POLarity?

**Arguments**  
NEGAtive specifies a negative pulse.
POSITIve specifies a positive pulse.

**TRIGger:MAIn:PULse:WIDth:WHEn**

*TDS 510A, 500B, 600B, & 700A Only*

Selects the condition when the trigger occurs. This is equivalent to selecting the condition in the Pulse Width Trig When side menu.

**Group** Trigger

**Syntax** TRIGger:MAIn:PULse:WIDth:WHEn { OUTside | WITHin }

TRIGger:MAIn:PULse:WIDth:WHEn?

**Arguments**

OUTside specifies a trigger when the duration of the pulse is greater than the high limit or less than the low limit specified. The high and low limits are specified with the TRIGger:MAIn:PULse:WIDth:HIGHLimit and TRIGger:MAIn:PULse:WIDth:LOWLimit commands respectively.

WITHin specifies a trigger when the duration of the pulse is within the high and low limits. The high and low limits are specified with the TRIGger:MAIn:PULse:WIDth:HIGHLimit and TRIGger:MAIn:PULse:WIDth:LOWLimit commands respectively.

**Examples** TRIGGER:MAIn:PULSE:WIDth:WHEN?

returns either OUTSIDE or WITHIN, indicating the conditions for generating a pulse trigger.

**TRIGger:MAIn:TYPe**

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

**Group** Trigger
**Syntax**

`TRIGger:MAIn:TPYe { EDGE | LOGIc | PULse | VIdeo }`

(Only the TDS 510A, 500B, 600B, & 700A use the LOGIc and PULse arguments. Only digitizing oscilloscopes with option 05 use the VIdeo argument.)

`TRIGger:MAIn:TPYe?`

**Arguments**

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

LOGIc (TDS 510A, 500B, 600B, & 700A only) specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:MAIn:LOGIc commands.

PULse (TDS 510A, 500B, 600B, & 700A only) specifies that a trigger occurs when a specified pulse is found and is controlled by the TRIGger:MAIn:PULse commands.

VIdeo (option 05 only) 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 PULSE indicating that the main trigger type is a pulse trigger.

---

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

*Option 05 Only*

Returns the main video trigger parameters.

**Group**

Trigger

**Syntax**

`TRIGger:MAIn:VIdeo?`
Examples

TRIGGER:MAIN:VIDEO?
might return: NTS;CH1;NEGA;NUMER;2;1;COLO;COLO;787;
59.94E+0;1050;2;890.0E-9;3.56E-6;15.00E-6;11.56E-6;
15.89E-6 as the current main video trigger parameters.

TRIGGER:MAIN:VIDEO:BY

_TDS 400A Option 05 Only_

Sets or queries the video trigger delay mode. This is equivalent to using the Video TV Delay Mode side menu.

Group Trigger

Syntax TRIGGER:MAIN:VIDEO:BY { TIME | LINES | LINE }
TRIGGER:MAIN:VIDEO:BY?

Arguments
TIME specifies a delay by time.
LINES specifies a delay by a number of video lines. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to LINE. The TDS 400A will not output this argument in response to a query.
LINE specifies a delay by a number of video lines.

Examples TRIGGER:MAIN:VIDEO:BY TIME
specifies a delay by time.

TRIGGER:MAIN:VIDEO:FIELD

_TDS 400A Option 05 Only_

Sets or queries the field the video trigger acts on. This is equivalent to using the Video Scan side menu when Class is NOT set to Custom.
**Group** Trigger

**Syntax**

```
TRIGger:MAIn:VIDeo:FIELD { ODD | EVEN | ALL | FIELD1 | FIELD2 | FIELDEither }
```

**Arguments**

- **ODD** specifies interlaced video field 1.
- **EVEN** specifies interlaced video field 2.
- **ALL** specifies alternating both video field 1 and video field 2.
- **FIELD1** specifies interlaced video field 1. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to ODD. The TDS 400A will not output this argument in response to a query.
- **FIELD2** specifies interlaced video field 2. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to EVEN. The TDS 400A will not output this argument in response to a query.
- **FIELDEither** specifies alternating both video field 1 and video field 2. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to ALL. The TDS 400A will not output this argument in response to a query.

**Examples**

```
TRIGGER:MAIN:VIDEO:FIELD ODD
```

selects odd fields.
**TRIGger:MAIn:VIDeo:FIELD**

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the field the video trigger acts on. This is equivalent to pressing **Field** in the video main menu, then **Field** in the side menu, and entering a value with the keypad or general purpose knob.

**Group**  
Trigger

**Syntax**

TRIGger:MAIn:VIDeo:FIELD <NR1>

TRIGger:MAIn:VIDeo:FIELD?

**Arguments**

<NR1> specifies the video (color) field. For example, 1 specifies interlaced video field one and 2 specifies interlaced video field two.

PAL signals have 1 to 8 fields, HDTV signals have 1 or 2, and FlexFormat supports 1 or 2.

**Examples**

TRIGGER:MAIN:VIDEO:FIELD 1
selects field 1.

**TRIGger:MAIn:VIDeo:FIELDType**

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the field the video trigger acts on. This is equivalent to pressing **Field** in the video main menu and then **Field, Odd, Even** or **All** in the side menu.

**Group**  
Trigger

**Syntax**

TRIGger:MAIn:VIDeo:FIELDType { NUMERic | ALL | EVEN | ODD }

TRIGger:MAIn:VIDeo:FIELDType?
NUMERic specifies a selected line in the selected field. If you send this command when the mode is MONO or SECAM, the digitizing oscilloscope will generate an SRQ.

ALL specifies a selected line in all fields.

EVEN specifies a selected line in even fields.

ODD specifies a selected line in odd fields.

Examples

TRIGGER:MAIN:VIDEO:FIELDTYPE ALL selects a selected line in all fields.

TRIGger:MAIn:VIDeo:FLEXformat? (Query Only)

TDS 510A, 500B, 600B, & 700A Option 05 Only

Returns the main flexible-format video trigger parameters.

Group Trigger

Syntax TRIGger:MAIn:VIDeo:FLEXformat?

Examples TRIGGER:MAIN:VIDEO:FLEXFORMAT?

might return: 59.94E+0;1050;2;890.0E-9;3.56E-6;15.00E-6;11.56E-6;
15.89E-6 as the flexible-format video trigger parameters.

TRIGger:MAIn:VIDeo:FLEXformat:FIELDRATE

TDS 510A, 500B, 600B, & 700A Option 05 Only

Sets or queries the flexible-format video frames per second (e.g. 59.94 frames per second for 1050 and 50 for 1250). This is equivalent to selecting Setup from
the video main menu (with **FlexFmt** as the **Standard**), **Field Rate** from the side menu, and entering a value with the keypad or the general purpose knob.

**Group** Trigger

**Syntax**

TRIGger:MAIn:VIDeo:FLEXformat:FIELDRATE <NR3>

TRIGger:MAIn:VIDeo:FLEXformat:FIELDRATE?

**Arguments** <NR3> the field rate.

**Examples**

TRIGGER:MAIN:VIDEO:FLEXFORMAT:FIELDRATE?

returns the specified field rate.

---

**TRIGger:MAIn:VIDeo:FLEXformat:FIELDS**

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the flexible-format video fields. This is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Fields** from the side menu, and entering the value with the keypad or the general purpose knob.

**Group** Trigger

**Syntax**

TRIGger:MAIn:VIDeo:FLEXformat:FIELDS <NR1>

TRIGger:MAIn:VIDeo:FLEXformat:FIELDS?

---
Arguments <NR1> the number of fields in the standard.

Examples TRIGGER:MAIN:VIDEO:FLEXformat:FIELD?

returns the number of fields in the format.

**TRIGGER:MAIN:VIDEO:FLEXformat:LINES**

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the flexible-format video lines in a frame. This is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Lines** from the side menu, and entering the value with the keypad or the general purpose knob.

Group Trigger

Syntax TRIGGER:MAIN:VIDEO:FLEXformat:LINES <NR1>

TRIGGER:MAIN:VIDEO:FLEXformat:LINES?

Arguments <NR3> the frame lines.

Examples TRIGGER:MAIN:VIDEO:FLEXFORMAT:LINES?

returns the specified number of lines.
**TRIGger:MAIn:VIDeo:FLEXformat:NEGSyncwidth**  
*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the flexible-format negative sync width. The HDTV horizontal sync is a tri-level sync. The first of the two consecutive sync pulses used is negative and the second is positive. The positive sync pulse starts on the rising edge of the negative sync. The two pulses have the same width such that specifying the negative pulse is only required. Setting the width is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Sync Width** from the side menu, and entering the value with the keypad or the general purpose knob.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:VIDeo:FLEXformat:NEGSyncwidth <NR3>
TRIGger:MAIn:VIDeo:FLEXformat:NEGSyncwidth?

**Arguments**  
<NR3> the negative sync width.

**Examples**  
TRIGGER:MAIN:VIDEO:FLEXFORMAT:NEGSYNCWIDTH?
returns the specified flexible-format negative sync width.

---

**TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime**  
*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the time from the positive edge of the tri-sync pulse for the last line in the selected field (t₀) to the leading edge (negative) of the first negative vertical sync pulse. This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V1 Start Time** in the side menu, and entering a value with the keypad or the general purpose knob.

**Group**  
Trigger
Syntax

```plaintext
TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime <NR3>
TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime?
```

Arguments

- `<NR3>` the v1 startime.

Examples

```
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STARTTIME?
returns the specified HDTV v1 startime.
```

**TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime**

*TDS 510A, 500B, 600B, & 700A Option 5 Only*

Sets or queries the time from $t_0$ to the trailing edge (positive) of the first negative vertical sync pulse. This is equivalent selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V1 Stop Time** in the side menu, and entering a value with the keypad or the general purpose knob.

Group

**Trigger**

Syntax

```plaintext
TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime <NR3>
TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime?
```

Arguments

- `<NR3>` the v1 stoptime.

Examples

```
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STOPTIME?
returns the specified v1 stoptime.
```
TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime
_TDS 510A, 500B, 600B, & 700A Option 05 Only_

Sets or queries the time from the positive edge of the tri-sync pulse for the last line in the selected field \( t_0 \) to the leading edge (positive) of the second vertical sync pulse. Note: the second pulse may be a negative pulse or the negative portion of a tri-sync pulse that is within the last line (usually located at the \( \frac{1}{2} \) line point). This is equivalent to selecting **Setup** from the video main menu (with FlexFmt as the **Standard**), pressing **V2 Start Time** in the side menu, and entering a value with the keypad or the general purpose knob.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime <NR3>  
TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime?

**Arguments**  
<NR3> the v2 startime.

**Examples**  
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STARTTIME?  
returns the specified v2 startime.

TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime
_TDS 510A, 500B, 600B, & 700A Option 05 Only_

Sets or queries the time from \( t_0 \) to the trailing edge (positive) of the second negative vertical sync pulse. This is equivalent selecting **Setup** from the video main menu (with FlexFmt as the **Standard**), pressing **V2 Stop Time** in the side menu, and entering a value with the keypad or the general purpose knob.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime <NR3>
TRIGger:MAIn:VIDeo:FLEXformat:V2ST0ptime?

Arguments  

<NR3> the v2 stoptime.

Examples  

TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STOPTIME?
returns the specified v2 stoptime.

TRIGger:MAIn:VIDeo:HDTv

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the high definition TV frame rate. This is equivalent to toggling HDTV from the Video main-menu Standard pop-up, pressing Format, and then selecting a frame rate from the side menu.

Group  

Trigger

Syntax  

TRIGger:MAIn:VIDeo:HDTv <NR1>
TRIGger:MAIn:VIDeo:HDTv?

Arguments  

<NR1> specifies the frame rate. More precisely, 787 selects a 787/59.94/2:1 format. 1050 selects a 1050/59.94/2:1 format. 1125 selects a 1125/60/2:1 format. 1250 selects a 1250/50/2:1 format.

Examples  

TRIGGER:MAIN:VIDEO:HDTV 787
specifies 787/59.94/2:1 format.
TRIGger:MAIn:VIDeo:HOLdoff? (Query Only)
*TDS 400A Option 05 Only*

Returns the video trigger holdoff value.

**Group**
Trigger

**Syntax**
TRIGger:MAIn:VIDeo:HOLdoff?

**Examples**
TRIGGER:MAIN:VIDEO:HOLDOFF?
might return :TRIGGER:MAIN:VIDEO:HOLDOFF:VALUE 0.

TRIGger:MAIn:VIDeo:HOLdoff:VALue
*TDS 400A Option 05 Only*

Sets or queries the video trigger holdoff value. This is equivalent to setting Holdoff in the Mode & Holdoff side menu of the video trigger menu.

**Group**
Trigger

**Syntax**
TRIGger:MAIn:VIDeo:HOLdoff:VALue <NR1>
TRIGger:MAIn:VIDeo:HOLdoff:VALue?

**Arguments**
<NR1> is from 0 to 100, and is a percent of the holdoff range.

**Examples**
TRIGGER:MAIN:HOLDOFF:VALUE 10
set the holdoff value to be 10% of the holdoff range.
TRIGger:MAIn:VIDeo:INTERLAcce

*TDS 400A Option 05 Only*

Sets or queries the video trigger interlace format. This is equivalent to setting Interlace in the Scan Rate and Interlace main menu of the video trigger menu when Class is set to Custom.

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:VIDeo:INTERLAcce { FIELD1 | FIELD2 | FIELDEither | OFF }

TRIGger:MAIn:VIDeo:INTERLAcce?

---

TRIGger:MAIn:VIDeo:LINE

*Option 05 Only*

Sets or queries the video trigger in terms of a number of lines. This is equivalent to pressing the Line (TV Delay Mode in the TDS 400A) item in the video main menu, pressing Line in the side menu (if needed), and entering a value with the keypad or the general purpose knob. The minimum line number is the starting line in the field. For PAL, field 1 (odd fields in the TDS 400A) has line 2 (1 in the TDS 400A) as the minimum, and field 2 (even fields in the TDS 400A) has line 315 (314 in the TDS 400A).

**Group**  
Trigger

**Syntax**  
TRIGger:MAIn:VIDeo:LINE <NR1>

TRIGger:MAIn:VIDeo:LINE?
Arguments <NR1> specifies a number of lines to delay by.

Examples TRIGGER:MAIN:VIDEO:LINES 5
selects 5 lines for the desired delay period.

TRIGGER:MAIN:VIDEO:LINES

*TDS 400A Option 05 Only*

Sets or queries the video trigger delay in terms of a number of lines. This is equivalent to entering data in the Line item in the Video TV Delay Mode side menu. This command is available for backwards compatibility.

**Group** Trigger

**Syntax**

TRIGGER:MAIN:VIDEO:LINES <NR3>

TRIGGER:MAIN:VIDEO:LINES?

Arguments <NR3> specifies a number of lines to delay by.

Examples TRIGGER:MAIN:VIDEO:LINES 5
selects 5 lines for the desired delay period.

TRIGGER:MAIN:VIDEO:NTSC

*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the NTSC mode selection. This is equivalent to choosing 525/NTSC in the video main-menu pop-up, Mode in the main menu, and a side menu item (NTSC or Mono).

**Group** Trigger
Syntax

TRIGger:MAIn:VIDeo:NTSc { MONo | COLOR }

TRIGger:MAIn:VIDeo:NTSc?

Arguments

MONo specifies mode for noncolor signals.
COLOR specifies mode for color field triggering and enables numeric triggering.

Examples

TRIGGER:MAIN:VIDEO:NTSC MONO
specifies numeric fields are invalid.

TRIGger:MAIn:VIDeo:PAL

TDS 510A, 500B, 600B, & 700A Option 05 Only

Sets or queries the PAL video trigger mode selection. This is equivalent to
toggling 625/PAL in the video main-menu pop-up, Mode in the main menu, and
a side menu item (PAL, Mono, or SECAM).

Group

Trigger

Syntax

TRIGger:MAIn:VIDeo:PAL { MONo | COLOR | SECAm }

TRIGger:MAIn:VIDeo:PAL?

Arguments

MONo specifies mode for non-color signals.
COLOR specifies mode for color field triggering and enables numeric triggering.
SECAM specifies mode for SECAM signals.

**Examples**
TRIGGER:MAIN:VIDEO:PAL MONO
specifies non-color PAL signals.

**TRIGGER:MAIN:VIDEO:SCAN**

*TDS 400A Option 05 Only*

Sets or queries the video trigger scan parameters. This is equivalent to using the Video Scan Parameters side menu.

**Group**
Trigger

**Syntax**
TRIGGER:MAIN:VIDEO:SCAN { RATE1 | RATE2 | RATE3 | RATE4 }
TRIGGER:MAIN:VIDEO:SCAN?

**Arguments**
RATE1 specifies a 15 to 20 kHz video line rate.
RATE2 specifies a 20 to 25 kHz video line rate.
RATE3 specifies a 25 to 35 kHz video line rate.
RATE4 specifies a 35 to 64 kHz video line rate.

**Examples**
TRIGGER:MAIN:VIDEO:SCAN RATE1
selects rate 1.

**TRIGGER:MAIN:VIDEO:SCANPeriod**

*TDS 400A Option 05 Only*

Sets or queries the video trigger scan period.
**TRIGger:MAIn:VIDeo:SOUrce**  
*Option 05 Only*

Sets or queries the source for the main video trigger. This is equivalent to selecting the Source in the video main menu and a desired channel from the side menu.

**Arguments**  
CH<x> specifies one of the input channels (CH1, CH2, CH3, or CH4; or CH1, CH2, AUX1, or AUX2 on the TDS 520B, 620B, 680B, 724A, & 782A).

**Examples**  
TRIGGER:MAIN:VIDEO:SOURce CH1
selects channel 1 as the source for the main video trigger.

**TRIGger:MAIn:VIDeo:STANdard**  
*TDS 510A, 500B, 600B, & 700A Option 05 Only*

Sets or queries the video trigger standard. This is equivalent to selecting the standard in the video Standard pop-up (525/NTSC, 625/PAL, HDTV, or FlexFmt).
**Syntax**

TRIGger:MAIn:VIDeo:STANdard \{ NTSc \ | \ PAL \ | \ HDTv \ | \ FLEXformat\}  
TRIGger:MAIn:VIDeo:STANdard?

**Arguments**

NTSc specifies the NTSC 525/59.94/2:1 standard.

PAL specifies the PAL 625/59.94/2:1 standard.

HDTv allows selection of the following HDTV formats: 787/59.94/1:1, 1050/59.94/2:1, 1050/59.94/2:1, 1125/60/2:1, 1250/50/2:1.

FLEXformat allows the user to specify the video parameters. The default table is 1050/59.94/2:1.

**Examples**

TRIGGER:MAIN:VIDEO:STANDARD NTSC  
selects the NTSC video format.

**TRIGger:MAIn:VIDeo:SYNc**

*Option 05 Only*

Sets or queries the video trigger sync polarity. This is equivalent to selecting Sync Polarity from the video main menu and a side-menu item (Neg Sync or Pos Sync).

**Group**

Trigger

**Syntax**

TRIGger:MAIn:VIDeo:SYNc \{ POSITIVe \ | \ NEGAtive \}  
TRIGger:MAIn:VIDeo:SYNc?
**Arguments**
POSITIVE specifies a positive going voltage.
NEGATIVE specifies a negative going voltage.

**Examples**
TRIGGER:MAIN:VIDEO:SYNC POSITIVE
selects a positive going voltage for the desired synchronization pulse.

---

**TRIGGER:MAIN:VIDEO:SYSTEM**

*TDS 400A Option 05 Only*

Sets or queries the video trigger class. This is equivalent to selecting the class in the Video Class side menu of the Video menu.

**Group**
Trigger

**Syntax**
TRIGGER:MAIN:VIDEO:SYSTEM { NTSC | PAL | SECAM | CUSTOM }
TRIGGER:MAIN:VIDEO:SYSTEM?

**Arguments**
NTSC selects a condition that adheres to the National Television System Committee standards. Specifically, it assumes a line rate of 525 lines per frame and a frame rate of 30 Hz.

PAL selects a condition that adheres to the Phase Alternate Line standard. Specifically, it assumes a line rate of 625 lines per frame and a frame rate of 25 Hz.

SECAM selects a condition that adheres to the SECAM standard.
CUSTom selects a condition that adheres to the frequency range of the video signal as you have defined them from the available ranges.

**Examples**

\[ \text{TRIGGER:MAIN:SYSTEM NTSC} \]

selects triggering to occur on an NTSC compatible signal.

**TRIGger:MAIn:VIDeo:TIMe**

_TDS 400A Option 05 Only_

Sets or queries the video trigger delay time. This is equivalent to entering the time in the _Delay by Time_ item of the Video _TV Delay Mode_ side menu.

**Group**

Trigger

**Syntax**

\[ \text{TRIGger:MAIn:VIDeo:TIMe <NR3>} \]

\[ \text{TRIGger:MAIn:VIDeo:TIMe?} \]

**Arguments**

<NR3> specifies a delay time.

**Examples**

\[ \text{TRIGGER:MAIN:VIDEO:TIME 5E–6} \]

selects 5 μs for the desired delay time.

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

Returns the current state of the triggering system.

**Group**

Trigger

**Syntax**

\[ \text{TRIGger:STATE?} \]
**Returns**

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

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

FASTframe (TDS 500B and 700A only) indicates that the instrument is in FastFrame mode. This means normal trigger status monitoring is turned off. The digitizing oscilloscope will not return armed, partial, ready, trigger, or auto while in this state.

INSTavu (TDS 500B and 700A only) indicates that the instrument is in InstaVu mode.

PARTial indicates that the main trigger has occurred and the digitizing oscilloscope is waiting for trigger(s) for the delay by events.

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

SAVE indicates that the digitizing oscilloscope is in save mode and is not acquiring data.

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

**Examples**

TRIGGER:STATE?

might return ARMED, indicating that pretrigger data is being acquired.

**TRG (No Query Form)**

(Trigger) executes commands that are defined by *DDT.

The Group Execute Trigger (GET) interface message has the same effect as the *TRG command.

**Group**

Miscellaneous

**Related Commands**

Alias commands, *DDT

**Syntax**

*TRG
**Examples**

*TRG

Immediately executes all commands that have been defined by *DDT.

---

**TST? (Query Only)**

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

<table>
<thead>
<tr>
<th><strong>Group</strong></th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntax</strong></td>
<td>*TST?</td>
</tr>
<tr>
<td><strong>Returns</strong></td>
<td>&lt;NR1&gt; and is always 0.</td>
</tr>
</tbody>
</table>

**UNLock (No Query Form)**

Unlocks the front panel. This command is equivalent to LOck NONE.

**NOTE.** If the digitizing oscilloscope is in the Remote With Lockout State (RWLS), the UNLOCK command has no effect. For more information see the ANSI-IEEE Std. 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

<table>
<thead>
<tr>
<th><strong>Group</strong></th>
<th>Miscellaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Related Commands</strong></td>
<td>LOck</td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td>UNLock ALL</td>
</tr>
<tr>
<td><strong>Arguments</strong></td>
<td>ALL specifies all front-panel buttons and knobs.</td>
</tr>
</tbody>
</table>
**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

**Related Commands**  
HEADer, *LRN?, SET?

**Syntax**  
VERBose { OFF | ON | <NR1> }

VERBose?

**Arguments**  
ON or <NR1> ≠ 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**  
VERBSE ON  
sets the Verbose State true.

VERBSE?  
might return the value 1, showing that the Verbose State is true.

**WAI (No Query Form)**

(Wait) Prevents the digitizing oscilloscope from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the digitizing oscilloscope with your application program. Synchronization methods are described on page 3–7.

**Group**  
Status and Error
**Command Descriptions**

**Related Commands**

BUSY?, *OPC

**Syntax**

*WAI

**WAVFrm? (Query Only)**

Returns WFMPre? and CURVe? data for the waveform or waveforms as specified by the DATa:SOUrce command. This command is equivalent to sending WFMPre?; CURVe?

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain 50 c1ks/div.

**Group**

Waveform

**Related Commands**

CURVe?, DATa:SOUrce, WFMPre?

**Syntax**

WAVFrm?

**WFMPre? (Query Only)**

Returns the waveform formatting data for the first ordered waveform as specified by the DATa:SOUrce command. The channel and math waveforms selected by the DATa:SOUrce command must be displayed.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain 50 c1ks/div.

**Group**

Waveform

**Related Commands**

WAVFrm?

**Syntax**

WFMPre?
Returns the number of bits per binary waveform point for the first ordered waveform as specified by the DATa:SOURce command. The WFMPre:BIT_Nr command is ignored on input.

Group Waveform

Related Commands DATa:WIDth, WFMPre:BYTE_Nr

Syntax WFMPre:BIT_Nr <NR1>

WFMPre:BIT_Nr?

Arguments <NR1> is either 8 or 16, and is equivalent to WFMPre:BYTE_Nr * 8.
**Examples**

WFMPre:BN_FMT might return 8, indicating that there are 8 bits per waveform point.

---

**WFMPre:BN_FMT**

Sets or queries the format of binary data for the first ordered waveform as specified by the DATa:SOURce command.

**Group**

Waveform

**Related Commands**

DATa:ENCdg, WFMPre:BYT_Nr, WFMPre:ENCdg

**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.

---

**WFMPre:BYT_Nr**

Sets or queries the binary field data width for the first ordered waveform as specified by the DATa:SOURce command. This command is equivalent to the DATa:WIDth command.

**Group**

Waveform

**Related Commands**

DATa:WIDth, WFMPre:BIT_Nr
Syntax  
WFMPre:BYT_Nr <NR1>
WFMPre:BYT_Nr?

Arguments  
<NR1> is the number of bytes per point and can be 1 or 2.

Examples  
WFMPRE:BYT_NR 2
specifies that there are 2 bytes per waveform data point.

WFMPre:BYT_Or

Selects 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.

Group  
Waveform

Related Commands  
DATa:ENCdg, WFMPre:BN_FMT, WFMPre:ENCdg

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 will be transferred first.
WFMPRE:BYT_OR?
returns either MSB or LSB depending on which data byte is transferred first.
**WFMPre:ENCdg**

Sets or queries the type of encoding for waveform data transferred with the CURVe command.

**Group**
Waveform

**Related Commands**
DATa:ENCdg, WFMPre:BYT_Or, WFMPre:BNFmt

**Syntax**
WFMPre:ENCdg { ASC | BIN }

**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.

---

**WFMPre:PT_Fmt (No Query Form)**

Selects the point format of the waveform data for the first ordered waveform as specified by the DATa:SOUrce command.

**Group**
Waveform

**Syntax**
WFMPre:PT_FMT { ENV | Y }

Arguments

ENV specifies that the waveform is transmitted as maximum and minimum point pairs. Only \( y \) values are explicitly transmitted. Absolute coordinates are given by:

\[
X_n = \text{XZEr}o + \text{XINc}r (n-\text{PT \_Off})
\]
\[
Y_{n\text{max}} = \text{YZEr}o + \text{YMUlt} (\text{y}_{n\text{max}} - \text{YO}ff)
\]
\[
Y_{n\text{min}} = \text{YZEr}o + \text{YMUlt} (\text{y}_{n\text{min}} - \text{YO}ff)
\]

\( Y \) 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:

\[
X_n = \text{XZEr}o + \text{XINc}r (n-\text{PT \_Off})
\]
\[
Y_n = \text{YZEr}o + \text{YMUlt} (\text{y}_n - \text{YO}ff)
\]

Examples

WFMPRE:PT \_Fmt \_ENV

sets the waveform data point format to enveloped.

**WFMPRE:PT \_Off (No Query Form)**

Specifies the trigger point within the waveform record for the reference waveform specified by the DATa:DESTination command.

**Group**

Waveform

**Related Commands**

HORizontal:TRIGger:POsition

**Syntax**

WFMPRE:PT \_Off <NR1>

**Arguments**

<NR1> = 0 to the recordlength, and is the position of the trigger point relative to DATa:STARt.

**Examples**

WFMPRE:PT \_OFF 1

specifies that the trigger point is the first point in the waveform record.
**WFMPre:XINcr (No Query Form)**

Specifies the horizontal sampling interval for the reference waveform specified by the DATa:DESTination command.

- **Group**: Waveform
- **Syntax**: WFMPre:XINcr <NR3>
- **Arguments**: <NR3> is the sampling interval in seconds per point.

**WFMPre:YMUlt (No Query Form)**

Specifies the vertical scale factor for the reference waveform specified by the DATa:DESTination command.

- **Group**: Waveform
- **Syntax**: WFMPre:YMUlt <NR3>
- **Arguments**: <NR3> is the vertical scale factor in YUunits (usually volts) per division.

**WFMPre:YOFF (No Query Form)**

Specifies the offset of the vertical component for the reference waveform specified by the DATa:DESTination command.

- **Group**: Waveform
- **Syntax**: WFMPre:YOFF <NR3>
**Arguments**  
<NR3> is the vertical offset in digitizing levels.

### WFMPre:YZEro (No Query Form)

Specifies the offset voltage for the reference waveform specified by the DATa:DESTination command.

**Group**  
Waveform

**Syntax**  
WFMPre:YZEro <NR3>

**Arguments**  
<NR3> is of the offset in YUNits (usually volts).

Table 2–31 lists additional WFMPre commands that are included for compatibility purposes.

**NOTE:** These commands do not support a query form and all information is ignored.

#### Table 2–31: Additional WFMPre Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMPre:CRVchk</td>
<td>(CHKsmo</td>
<td>NONE)</td>
</tr>
<tr>
<td>WFMPre:NR_PT</td>
<td>&lt;NR1&gt;</td>
<td>Number of waveform points</td>
</tr>
<tr>
<td>WFMPre:WFId</td>
<td>&lt;QSTRING&gt;</td>
<td>Waveform identifier</td>
</tr>
<tr>
<td>WFMPre:XUNIT</td>
<td>&lt;QSTRING&gt;</td>
<td>Horizontal units</td>
</tr>
<tr>
<td>WFMPre:XMULT</td>
<td>&lt;NR3&gt;</td>
<td>Horizontal (X-axis) scale factor</td>
</tr>
<tr>
<td>WFMPre:XOFF</td>
<td>&lt;NR3&gt;</td>
<td>Horizontal (X-axis) offset</td>
</tr>
<tr>
<td>WFMPre:YUNIT</td>
<td>&lt;QSTRING&gt;</td>
<td>Vertical units</td>
</tr>
<tr>
<td>WFMPre:ZMULT</td>
<td>&lt;NR3&gt;</td>
<td>Z-axis scale factor</td>
</tr>
<tr>
<td>WFMPre:ZOFF</td>
<td>&lt;NR3&gt;</td>
<td>Z-axis offset</td>
</tr>
<tr>
<td>WFMPre:ZUNIT</td>
<td>&lt;QSTRING&gt;</td>
<td>Z-axis units</td>
</tr>
<tr>
<td>WFMPre:ZZEro</td>
<td>&lt;NR3&gt;</td>
<td>Z-axis origin offset</td>
</tr>
</tbody>
</table>
NOTE. When returning WFMPRE:<wfm> information from the oscilloscope, <wfm> specifies the waveform source (CH<x>, MATH<x>, or REF<x>). The source must also be set using the DA:SOUrce command. When sending WFMPRE:<wfm> information to the scope, the <wfm> specification is ignored and the reference location specified by DA:DESTination is used instead.

WFMPre:<wfm>? (Query Only)

Returns the waveform formatting data for first ordered waveform as specified by the DA:SOUrce command. Channel and math waveforms must be displayed before they can be queried. Querying an invalid reference waveform generates an execution error.

Group    Waveform

Syntax    WFMPRE:<wfm>?

Returns    The format of the response is:

<wfm>:WID <QString>; NR_PT <NR1>; PT_FMT { ENV | Y }; XUNIT <QString>; XIncr <NR3>; PT_0FF <NR1>; YUNIT <QString>; YMULT <NR3>; YOFF <NR3>; YZERo <NR3>

[; <wfm>:WID <QString>; NR_PT <NR1>; PT_FMT { ENV | Y }; XUNIT <QString>; XIncr <NR3>; XZERo <NR3>; PT_0FF <NR1>; YUNIT <QString>; YMULT <NR3>; YOFF <NR3>; YZERo <NR3>...]

Example    WFMPRE:CH1?

might return

:WFMPRE:CH1:WID "Ch1, DC coupling, 100.0mVolts/div, 500.0us/div, 500 points, Sample mode"; NR_PT 1; PT_FMT Y; XUNIT "s"; XINCR 10.00E-6; XZERo 8.15E-6; PT_0FF -249; YUNIT "Volts"; YMULT 4.000E-3; YOFF 0.0E+0; YZERo 0.0E+0C

WFMPre:<wfm>:NR_Pt

Sets or queries the number of points that are in the transmitted waveform record. This value is ignored on input.
Related Commands  
**DATa:DESTination**

**Group**  
Waveform

**Syntax**  
WFMPRe:<wfm>:PT_Fmt <NR1>

WFMPRe:<wfm>:PT?  

**Arguments**  
<NR1> is the number of data points. If DATa:WIDth is 2 then there are twice as many bytes.

<NR1> = 0 means that the waveform record is of an unspecified length.

**Examples**  
WFMPRe:CH1:NR_Pt?

might return 5000 as the number of data points in the waveform record transferred from channel 1.

**WFMPRe:<wfm>:PT_Fmt**  

Selects the data point format for the first ordered waveform as selected by the DATa:SOURce command. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

**Group**  
Waveform

**Related Commands**  
DATa:DESTination

**Syntax**  
WFMPRe:<wfm>:PT_Fmt { ENV | Y }

WFMPRe:<wfm>:PT_Fmt?
Arguments  ENV specifies that the waveform is transmitted as minimum and maximum point pairs. Only y values are explicitly transmitted. Absolute coordinates are given by:

\[
X_n = XZ\text{E}ro + XINcr (n–PT\_Off)
\]
\[
Y_{\text{min}} = YZ\text{E}ro + YMUlt (y_{\text{min}} - YOF)
\]
\[
Y_{\text{max}} = YZ\text{E}ro + YMUlt (y_{\text{max}} - YOF)
\]

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:

\[
X_n = XZ\text{E}ro + XINcr (n–PT\_Off)
\]
\[
Y_n = YZ\text{E}ro + YMUlt (y_n - YOF)
\]

Examples  WFMPRE:MATH1:PT_FMT?

might return ENV, indicating that the MATH1 waveform data format is enveloped.

WFMPRE:<wfm>:PT_Off

Returns the trigger point within the waveform record. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group  Waveform

Syntax  WFMPRE:<wfm>:PT_Off <NR1>

WFMPRE:<wfm>:PT_Off?

Arguments  <NR1> = 0 to the recordlength, and is the position of the trigger point relative to DATa:STARt when queried.

Examples  WFMPRE:CH1:PT_OFF?

returns 0 indicating the trigger position within the waveform record.
**WFMPre:<wfm>:WFId**

Returns information about the waveform such as input coupling, volts per division, time per division, acquisition mode, and record length.

The WFMPre:<wfm>:WFId command is ignored on input.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain “50 clks/div”.

**Group**
Waveform

**Syntax**
WFMPre:<wfm>:WFId <QString>
WFMPre:<wfm>:WFId?

**Arguments**
<QString> is the waveform identifier string.

**WFMPre:<wfm>:XINcr**

Sets or queries the horizontal sampling interval. On input <wfm> always defaults to the reference location specified by DATa:DESTinationation regardless of what is sent.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain “50 clks/div”.

**Group**
Waveform

**Syntax**
WFMPre:<wfm>:XINcr <NR3>
WFMPre:<wfm>:XINcr?

**Arguments**
<NR3> is the sampling interval.
WFMPre:<wfm>:XUNit

Returns the horizontal (X-axis) units of the waveform data at the time of creation.

The WFMPre:<wfm>:XUNit command is ignored on input.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain “50 clks/div”.

Group  Waveform

Syntax  WFMPre:<wfm>:XUNit <QString>
        WFMPre:<wfm>:XUNit?

Arguments  <QString> is “s” for seconds and specifies the units.

Examples  WFMPre:CH1:XUNIT?
          might return "s", indicating that the horizontal units for channel 1 are seconds.

WFMPre:<wfm>:XZErO

*TDS 500B, 600B, & 700A Only*

Sets or queries the horizontal (X-axis) origin offset. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group  Waveform

Syntax  WFMPre:<wfm>:XZErO <NR3>
        WFMPre:<wfm>:XZErO?
Command Descriptions

**WFMPre:<wfm>:YMUlt**

Sets or queries the vertical scale factor, in YUNit per unscaled data point value. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

**Group** Waveform

**Syntax**

```
WFMPre:<wfm>:YMUlt <NR3>
WFMPre:<wfm>:YMUlt?
```

**Arguments** <NR3> is the scale factor, in YUNits (usually volts), per digitizing level.

**WFMPre:<wfm>:YOFf**

Sets or queries the vertical position of the waveform. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

**Group** Waveform

**Syntax**

```
WFMPre:<wfm>:YOFf <NR3>
WFMPre:<wfm>:YOFf?
```

**Arguments** <NR3> is the offset in XUNits (usually time).
Arguments <NR3> is the position in digitizing levels.

**WFMPRe:<wfm>:YUNit**

Returns the vertical (Y-axis) units of the waveform data at the time of creation. The WFMPRe:<wfm>:YUNit command is ignored on input.

**Group** Waveform

**Syntax** WFMPRe:<wfm>:YUNit <QString>
WFMPRe:<wfm>:YUNit?

Arguments <QString> is "V" for volts or "VV" for volts², and specifies the units.

Examples WFMPRE:CH2:YUNIT?
might return "V", meaning that the units for the vertical component of the channel 2 waveform data are volts.

**WFMPRe:<wfm>:YZEro**

Sets or queries the vertical (Y-axis) offset voltage. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

**Group** Waveform

**Syntax** WFMPRe:<wfm>:YZEro <NR3>
WFMPRe:<wfm>:YZEro?


Arguments

<NR3> is the offset in YUnits (usually volts).

**ZOOM**

Resets the zoom transforms to default values for all traces or live traces. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display. This command is equivalent to selecting Reset All Factors or Reset Live Factors in the TDS 400A, 500B, 600B, 700A Zoom Reset menu or selecting Reset Zoom Factors in the Zoom menu of the TDS 510A.

**Group**

Zoom

**Syntax**

```
ZOOM { RESET | RESETLive (TDS 400A, 500B, 600B, and 700A only) }
```

```
ZOOM?
```

**Arguments**

RESET resets the zoom transforms to default values for all traces.

RESETLive (TDS 400A, 500B, 600B, and 700A only) resets the zoom transforms to default values for live traces.

**Examples**

```
ZOOM? might return :ZOOM:STATE OFF;HORIZONTAL:SCALE 2.00E+0;POSITION 500.0E-3;LOCK LIVE;:ZOOM:VERTICAL:SCALE 2.0E+0;POSITION 0.0E+0.
```

**ZOOM:DUAL**

*TDS 400A, 500B, 600B, & 700A Only*

Turns Dual Zoom mode on and off.

**Group**

Zoom

**Syntax**

```
ZOOM:DUAL { OFF | ON | <NR1> }
```

```
ZOOM:DUAL?
```
Arguments 0FF or <NR1> = 0 turns Dual Zoom mode off.
ON or <NR1> ≠ 0 turns Dual Zoom mode on.

Examples ZOOM: DUAL ON enables the Dual Zoom feature.
ZOOM: DUAL? returns either 0 or 1 depending on the state of Dual Zoom mode.

**ZOOM: DUAL: OFFSET**

*TDS 400A, 500B, 600B, & 700A Only*

Adjusts the requested horizontal offset between the centers of the main and second zoom boxes.

**Group**  Zoom

**Syntax**  ZOOM: DUAL: OFFSET <NR3>
ZOOM: DUAL: OFFSET?

Arguments  <NR3> is the offset time in seconds.

Examples  ZOOM: DUAL: OFFSET 100.0E-6 adjusts the offset time in seconds between the centers of the main and second zoom boxes.
**ZOOm:GRAticule**

*TDS 400A, 500B, 600B, & 700A Only*

Selects between the upper and lower graticule for use by the zoom preview state. If you select the lower graticule, horizontal and vertical knob changes will affect the underlying acquisition system. If you select the upper graticule, horizontal and vertical knob changes will affect the zoom factors.

**Group**

Zoom

**Syntax**

ZOOm:GRAticule { L0WEr | UPPer }

ZOOm:GRAticule?

**Arguments**

L0WEr selects the lower graticule.

UPPer selects the upper graticule.

**Examples**

ZOOm:GRAticule?

might return UPPer, indicating that the upper graticule is selected.

**ZOOm:HORizontal:LOCk**

Specifies the waveforms that the horizontal zoom parameters affect. This is equivalent to setting Horizonal Lock in the Zoom side menu.

**Group**

Zoom

**Syntax**

ZOOm:HORizontal:LOCk { ALL | LIVe | NONE }

ZOOm:HORizontal:LOCk?
ZOOm:HORizontal:POSition

Sets or queries the horizontal position of zoomed waveforms. The setting of the ZOOm:HORizontal:LOCk command determines the waveforms affected. For example, if ZOOm:HORizontal:LOCk is set to LIVe then only live (as opposed to reference and math) waveforms are affected.

**Group**
Zoom

**Syntax**
ZOOm:HORizontal:POSition <NR3>
ZOOm:HORizontal:POSition?

**Arguments**

<NR3> is from 0 to 100, and is the percent of the waveform that is to the left of screen center when the zoom factor is 1x or greater.
It is the percent of the graticule to the right of the waveform center when the zoom factor is less than 1. It can be less than 1 when using Fit to Screen.

**Examples**

```
ZOOM:HORIZONTAL:POSITION 50
```

centers the waveform on the display.

### ZOOM:HORIZONTAL:SCALE

Sets or queries the horizontal expansion factor. This command is equivalent to using the front-panel **Horizontal Scale** knob when Zoom is on. It is also equivalent to using **Zoom Preview** with the upper graticule selected in the TDS 500B, 600B or 700A.

**Group**

Zoom

**Syntax**

```
ZOOM:HORIZONTAL:SCALE <NR3>
```

```
ZOOM:HORIZONTAL:SCALE?
```

**Arguments**

<NR3> is the amount of expansion in the horizontal direction.

- It is less than 1 if there is horizontal compression.
- It is more than 1 if there is horizontal expansion.

**Examples**

```
ZOOM:HORIZONTAL:SCALE?
```

might return 1.00E+0 as the horizontal scale factor.

### ZOOM:STATE

Turns Zoom mode on and off. When Zoom mode is on, the horizontal and vertical position and scale commands affect the waveform display, not the acquisition. This is the only way to position and scale math and reference waveforms. This command is equivalent to turning **Zoom** on and off in the Zoom side menu.
**Group**  
Zoom

**Syntax**  
ZOOM:STATE { OFF | ON | PREview }
ZOOM:STATE?

**Arguments**  
OFF turns Zoom mode off.

ON turns Zoom mode on. When InstaVu is in use, the state value becomes a requested value, to be restored when InstaVu is switched off. Do not use this condition if InstaVu is on. InstaVu overrides the zoom on state.

PREview sets ZOOM:STATE to preview. Also, it causes the oscilloscope to display both the ZOOM:STATE OFF and ZOOM:STATE ON traces simultaneously in dual, half-height graticules. Do not use this condition if InstaVu is on. InstaVu overrides the zoom preview state.

**Examples**  
ZOOM:STATE ON
enables the Zoom feature.

ZOOM:STATE?
returns OFF, PREVIEW, or ON depending on the state of Zoom mode.

**ZOOM:VERTical:POSITION**

Sets or queries the vertical position of waveforms.

**Group**  
Zoom

**Syntax**  
ZOOM:VERTical:POSITION <NR3>
ZOOM:VERTical:POSITION?
Command Descriptions

ZOOM:VERTical:POSITION

Sets or queries the vertical expansion and compression factor.

Arguments  

<NR3> is the vertical position in divisions.

Examples  

ZOOM:VERTICAL:POSITION?  

might return :ZOOM:VERTICAL:POSITION 0.0E+0

ZOOM:VERTical:SCALE

Sets or queries the vertical expansion and compression factor.

Arguments  

<NR3> is the amount of vertical expansion or compression.

Examples  

ZOOM:VERTICAL:SCALE?  

might return :ZOOM:VERTICAL:SCALE 2.0E+0
Status and Events

The digitizing oscilloscope provides a status and event reporting system for the GPIB interface. This system informs you of certain significant events that occur within the digitizing oscilloscope.

The digitizing oscilloscope status handling system consists of five 8-bit registers and two queues. This section describes these registers and components. It also explains how the event handling system operates.

Registers

The registers in the event handling system fall into two functional groups:

- **Status Registers** contain information about the status of the digitizing oscilloscope. They include the Standard Event Status Register (SESR) and the Status Byte Register (SBR).

- **Enable Registers** determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

**Status Registers**

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the digitizing 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 digitizing 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 3–3.) 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 digitizing oscilloscope. Use the *ESR?* query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.
Figure 3–1: The Standard Event Status Register (SESR)

Table 3–1: SESR Bit Functions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 (MSB)</td>
<td>PON (Power On). Shows that the digitizing oscilloscope was powered on. The completion of the diagnostic tests also sets this bit.</td>
</tr>
<tr>
<td>6</td>
<td>URQ (User Request). Shows that an Application menu button was pressed.</td>
</tr>
<tr>
<td>5</td>
<td>CME (Command Error). Shows that an error occurred while the digitizing oscilloscope was parsing a command or query. Command error messages are listed in Table 3–4 on page 3–12.</td>
</tr>
<tr>
<td>4</td>
<td>EXE (Execution Error). Shows that an error occurred while the digitizing oscilloscope was executing a command or query. Execution error messages are listed in Table 3–5 on page 3–14.</td>
</tr>
<tr>
<td>3</td>
<td>DDE (Device Error). Shows that a device error occurred. Device error messages are listed in Table 3–6 on page 3–17.</td>
</tr>
<tr>
<td>2</td>
<td>QYE (Query Error). Shows that either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.</td>
</tr>
<tr>
<td>1</td>
<td>RQC (Request Control). Not used.</td>
</tr>
<tr>
<td>0 (LSB)</td>
<td>OPC (Operation Complete). Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.</td>
</tr>
</tbody>
</table>

The Status Byte Register (SBR) — shown in Figure 3–2, records whether output is available in the Output Queue, whether the digitizing oscilloscope requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

Figure 3–2: The Status Byte Register (SBR)
Table 3–2: SBR Bit Functions

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 (MSB)</td>
<td>Not used.</td>
</tr>
<tr>
<td>6</td>
<td><strong>RQS</strong> (Request Service), obtained from a serial poll. Shows that the digitizing oscilloscope requests service from the GPIB controller.</td>
</tr>
<tr>
<td>6</td>
<td><strong>MSS</strong> (Master Status Summary), obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.</td>
</tr>
<tr>
<td>5</td>
<td><strong>ESB</strong> (Event Status Bit). Shows that status is enabled and present in the SESR.</td>
</tr>
<tr>
<td>4</td>
<td><strong>MAV</strong> (Message Available). Shows that output is available in the Output Queue.</td>
</tr>
<tr>
<td>3 – 0</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to 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.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER) — is shown in Figure 3–3. This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR, as described earlier.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

![Figure 3–3: The Device Event Status Enable Register (DESER)](image)

The Event Status Enable Register (ESER) — is shown in Figure 3–4. It controls which types of events are summarized by the Event Status Bit (ESB) in the SBR.

Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.
The Service Request Enable Register (SRER) — is shown in Figure 3–5. It controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read it. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

The Enable Registers and the *PSC Command

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending *PSC 0 lets the Enable Registers maintain their values in non-volatile memory through a power cycle.

**NOTE.** To enable the PON (Power On) event to generate a Service Request, send *PSC 0, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.
Queues

The digitizing oscilloscope status and event reporting system contains two queues: the Output Queue and the Event Queue.

**The Output Queue**

The digitizing oscilloscope stores query responses in the Output Queue. It empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.

**WARNING.** When a controller sends a query, an <EOM>, and a second query, the digitizing scope normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

**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 the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.
Event Handling Sequence

Figure 3–6, on page 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 is generated (7).

**Synchronization Methods**

Although most GPIB commands are completed almost immediately after being received by the digitizing 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 digitizing oscilloscope will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one gets processed. The digitizing oscilloscope status and event reporting system provide ways to do this.

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```plaintext
/** Set up single-sequence acquisition **/
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE NORMALSAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/** Take amplitude measurement on acquired data **/
MEASUREMENT:IMMED:VALUE?
```

The acquisition of the waveform requires extended processing time. It may not finish before the digitizing oscilloscope takes an amplitude measurement (See Figure 3–7). This can result in an incorrect amplitude value.
To ensure the digitizing oscilloscope completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program. Figure 3–8 shows the desired processing sequence.

You can use four commands to synchronize the operation of the digitizing 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 looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1  ON
HORIZONTAL:RECORDLENGTH  500
ACQUIRE:MODE  NORMALSAMPLE
ACQUIRE:STOPAFTER  SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE  ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE  AMPLITUDE
MEASUREMENT:IMMED:SOURCE  CH1
/* Wait until the acquisition is complete before taking the measurement */
*WAI
/* Take amplitude measurement 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 digitizing oscilloscope is slowed since it is processing a single command at a time. This time could be spent doing other tasks.

The controller can continue to write commands to the input buffer of the digitizing oscilloscope, but the commands will not be processed by the digitizing oscilloscope until all operations in process are complete. If the input buffer becomes full, the controller will be unable to write more commands to the buffer. This can cause a time-out.

Using the BUSY Query

The BUSY? query allows you to find out whether the digitizing oscilloscope is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence using the BUSY? query for synchronization looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE NORMALSAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
While BUSY? keep looping
/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?

This sequence lets you create your own wait loop rather than using the *WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up, though, and the repeated BUSY? query will result in more bus traffic.

Using the *OPC Command

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.
**Serial Poll Method.** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands. When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement. */
*OPC
While serial poll = 0, keep looping
/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?

This technique requires less bus traffic than did looping on BUSY?.

**Service Request Method.** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands. You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, a Service Request will be generated.

The same command sequence using the *OPC command for synchronization looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
*OPC
Program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.
/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?

This technique is more efficient but requires more sophisticated programming.

**Using the *OPC? Query**

The *OPC? query 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 the *OPC? query for synchronization looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
*OPC?
Wait for read from Output Queue.
/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.
Messages

Tables 3–3 through 3–9 list all the programming interface messages the digitizing oscilloscope generates in response to commands and queries.

For most messages, a secondary message from the digitizing oscilloscope gives more detail about the cause of the error or the meaning of the message. This message is part of the message string, and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables 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

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No events to report – queue empty</td>
</tr>
<tr>
<td>1</td>
<td>No events to report – new events pending *ESR?</td>
</tr>
</tbody>
</table>

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 Command Syntax starting on page 2–1.

Table 3–4: Command Error Messages – CME Bit 5

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Command error</td>
</tr>
<tr>
<td>101</td>
<td>Invalid character</td>
</tr>
<tr>
<td>102</td>
<td>Syntax error</td>
</tr>
<tr>
<td>103</td>
<td>Invalid separator</td>
</tr>
<tr>
<td>104</td>
<td>Data type error</td>
</tr>
<tr>
<td>105</td>
<td>GET not allowed</td>
</tr>
<tr>
<td>106</td>
<td>Invalid program data separator</td>
</tr>
<tr>
<td>108</td>
<td>Parameter not allowed</td>
</tr>
<tr>
<td>109</td>
<td>Missing parameter</td>
</tr>
<tr>
<td>110</td>
<td>Command header error</td>
</tr>
</tbody>
</table>
### Table 3-4: Command Error Messages – CME Bit 5 (Cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Header separator error</td>
</tr>
<tr>
<td>112</td>
<td>Program mnemonic too long</td>
</tr>
<tr>
<td>113</td>
<td>Undefined header</td>
</tr>
<tr>
<td>118</td>
<td>Query not allowed</td>
</tr>
<tr>
<td>120</td>
<td>Numeric data error</td>
</tr>
<tr>
<td>121</td>
<td>Invalid character in number</td>
</tr>
<tr>
<td>123</td>
<td>Exponent too large</td>
</tr>
<tr>
<td>124</td>
<td>Too many digits</td>
</tr>
<tr>
<td>128</td>
<td>Numeric data not allowed</td>
</tr>
<tr>
<td>130</td>
<td>Suffix error</td>
</tr>
<tr>
<td>131</td>
<td>Invalid suffix</td>
</tr>
<tr>
<td>134</td>
<td>Suffix too long</td>
</tr>
<tr>
<td>138</td>
<td>Suffix not allowed</td>
</tr>
<tr>
<td>140</td>
<td>Character data error</td>
</tr>
<tr>
<td>141</td>
<td>Invalid character data</td>
</tr>
<tr>
<td>144</td>
<td>Character data too long</td>
</tr>
<tr>
<td>148</td>
<td>Character data not allowed</td>
</tr>
<tr>
<td>150</td>
<td>String data error</td>
</tr>
<tr>
<td>151</td>
<td>Invalid string data</td>
</tr>
<tr>
<td>152</td>
<td>String data too long</td>
</tr>
<tr>
<td>158</td>
<td>String data not allowed</td>
</tr>
<tr>
<td>160</td>
<td>Block data error</td>
</tr>
<tr>
<td>161</td>
<td>Invalid block data</td>
</tr>
<tr>
<td>168</td>
<td>Block data not allowed</td>
</tr>
<tr>
<td>170</td>
<td>Expression error</td>
</tr>
<tr>
<td>171</td>
<td>Invalid expression</td>
</tr>
<tr>
<td>178</td>
<td>Expression data not allowed</td>
</tr>
<tr>
<td>180</td>
<td>Alias error</td>
</tr>
<tr>
<td>181</td>
<td>Invalid outside alias definition</td>
</tr>
<tr>
<td>183</td>
<td>Invalid inside alias definition</td>
</tr>
<tr>
<td>184</td>
<td>Command in alias requires more/fewer parameters</td>
</tr>
</tbody>
</table>
Table 3–5 lists the execution 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**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Execution error</td>
</tr>
<tr>
<td>201</td>
<td>Invalid while in local</td>
</tr>
<tr>
<td>202</td>
<td>Settings lost due to rtl</td>
</tr>
<tr>
<td>210</td>
<td>Trigger error</td>
</tr>
<tr>
<td>211</td>
<td>Trigger ignored</td>
</tr>
<tr>
<td>212</td>
<td>Arm ignored</td>
</tr>
<tr>
<td>220</td>
<td>Parameter error</td>
</tr>
<tr>
<td>221</td>
<td>Settings conflict</td>
</tr>
<tr>
<td>222</td>
<td>Data out of range</td>
</tr>
<tr>
<td>223</td>
<td>Too much data</td>
</tr>
<tr>
<td>224</td>
<td>Illegal parameter value</td>
</tr>
<tr>
<td>230</td>
<td>Data corrupt or stale</td>
</tr>
<tr>
<td>240</td>
<td>Hardware error</td>
</tr>
<tr>
<td>241</td>
<td>Hardware missing</td>
</tr>
<tr>
<td>242</td>
<td>Hardware configuration error</td>
</tr>
<tr>
<td>243</td>
<td>Hardware I/O device error</td>
</tr>
<tr>
<td>250</td>
<td>Mass storage error</td>
</tr>
<tr>
<td>251</td>
<td>Missing mass storage</td>
</tr>
<tr>
<td>252</td>
<td>Missing media</td>
</tr>
<tr>
<td>253</td>
<td>Corrupt media</td>
</tr>
<tr>
<td>254</td>
<td>Media full</td>
</tr>
<tr>
<td>255</td>
<td>Directory full</td>
</tr>
<tr>
<td>256</td>
<td>File name not found</td>
</tr>
<tr>
<td>257</td>
<td>File name error</td>
</tr>
<tr>
<td>258</td>
<td>Media protected</td>
</tr>
<tr>
<td>260</td>
<td>Expression error</td>
</tr>
<tr>
<td>261</td>
<td>Math error in expression</td>
</tr>
<tr>
<td>2200</td>
<td>Measurement error, Measurement system error</td>
</tr>
<tr>
<td>2201</td>
<td>Measurement error, Zero period</td>
</tr>
<tr>
<td>2202</td>
<td>Measurement error, No period found</td>
</tr>
</tbody>
</table>
### Table 3–5: Execution Error Messages – EXE Bit 4 (Cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2203</td>
<td>Measurement error, No period, second waveform</td>
</tr>
<tr>
<td>2204</td>
<td>Measurement error, Low signal amplitude</td>
</tr>
<tr>
<td>2205</td>
<td>Measurement error, Low amplitude, second waveform</td>
</tr>
<tr>
<td>2206</td>
<td>Measurement error, Invalid gate</td>
</tr>
<tr>
<td>2207</td>
<td>Measurement error, Measurement overflow</td>
</tr>
<tr>
<td>2208</td>
<td>Measurement error, Waveform does not cross Mid Ref</td>
</tr>
<tr>
<td>2209</td>
<td>Measurement error, No second Mid Ref crossing</td>
</tr>
<tr>
<td>2210</td>
<td>Measurement error, No Mid Ref crossing, second waveform</td>
</tr>
<tr>
<td>2211</td>
<td>Measurement error, No backwards Mid Ref crossing</td>
</tr>
<tr>
<td>2212</td>
<td>Measurement error, No negative crossing</td>
</tr>
<tr>
<td>2213</td>
<td>Measurement error, No positive crossing</td>
</tr>
<tr>
<td>2214</td>
<td>Measurement error, No crossing</td>
</tr>
<tr>
<td>2215</td>
<td>Measurement error, No crossing, second waveform</td>
</tr>
<tr>
<td>2216</td>
<td>Measurement error, No crossing, target waveform</td>
</tr>
<tr>
<td>2217</td>
<td>Measurement error, Constant waveform</td>
</tr>
<tr>
<td>2218</td>
<td>Measurement error, Unused</td>
</tr>
<tr>
<td>2219</td>
<td>Measurement error, No valid edge – No arm sample</td>
</tr>
<tr>
<td>2220</td>
<td>Measurement error, No valid edge – No arm cross</td>
</tr>
<tr>
<td>2221</td>
<td>Measurement error, No valid edge – No trigger cross</td>
</tr>
<tr>
<td>2222</td>
<td>Measurement error, No valid edge – No second cross</td>
</tr>
<tr>
<td>2223</td>
<td>Measurement error, Waveform mismatch</td>
</tr>
<tr>
<td>2224</td>
<td>Measurement error, WAIT calculating</td>
</tr>
<tr>
<td>2225</td>
<td>Measurement error, No waveform to measure</td>
</tr>
<tr>
<td>2226</td>
<td>Null Waveform</td>
</tr>
<tr>
<td>2227</td>
<td>Positive and Negative Clipping</td>
</tr>
<tr>
<td>2228</td>
<td>Measurement error, Positive Clipping</td>
</tr>
<tr>
<td>2229</td>
<td>Measurement error, Negative Clipping</td>
</tr>
<tr>
<td>2230</td>
<td>Measurement error, High Ref &lt; Low Ref</td>
</tr>
<tr>
<td>2235</td>
<td>Math error, Invalid math description</td>
</tr>
<tr>
<td>2236</td>
<td>Math error, Reference waveform is invalid</td>
</tr>
<tr>
<td>2237</td>
<td>Out of acquisition memory</td>
</tr>
<tr>
<td>2238</td>
<td>Too many channels need to be acquired</td>
</tr>
<tr>
<td>2240</td>
<td>Invalid password</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2241</td>
<td>Waveform requested is invalid</td>
</tr>
<tr>
<td>2242</td>
<td>Data start and stop &gt; record length</td>
</tr>
<tr>
<td>2243</td>
<td>Waveform requested is not a data source</td>
</tr>
<tr>
<td>2244</td>
<td>Waveform requested is not turned on</td>
</tr>
<tr>
<td>2245</td>
<td>Saveref error, Selected channel is turned off</td>
</tr>
<tr>
<td>2246</td>
<td>Saveref error, Selected channel data invalid</td>
</tr>
<tr>
<td>2247</td>
<td>Saveref error, Out of reference memory</td>
</tr>
<tr>
<td>2248</td>
<td>Saveref error, Source reference data invalid</td>
</tr>
<tr>
<td>2249</td>
<td>Reference deletion error, Waveform in use for math</td>
</tr>
<tr>
<td>2260</td>
<td>Calibration error</td>
</tr>
<tr>
<td>2270</td>
<td>Alias error</td>
</tr>
<tr>
<td>2271</td>
<td>Alias syntax error</td>
</tr>
<tr>
<td>2272</td>
<td>Alias execution error</td>
</tr>
<tr>
<td>2273</td>
<td>Illegal alias label</td>
</tr>
<tr>
<td>2274</td>
<td>Alias parameter error</td>
</tr>
<tr>
<td>2275</td>
<td>Alias definition too long</td>
</tr>
<tr>
<td>2276</td>
<td>Alias expansion error</td>
</tr>
<tr>
<td>2277</td>
<td>Alias redefinition not allowed</td>
</tr>
<tr>
<td>2278</td>
<td>Alias header not found</td>
</tr>
<tr>
<td>2279</td>
<td>Alias label too long</td>
</tr>
<tr>
<td>2280</td>
<td>Alias table full</td>
</tr>
<tr>
<td>2285</td>
<td>TekSecure® Pass</td>
</tr>
<tr>
<td>2286</td>
<td>TekSecure® Fail</td>
</tr>
<tr>
<td>2290</td>
<td>Limit error, Reference in use</td>
</tr>
<tr>
<td>2291</td>
<td>Limit error, Reference data invalid</td>
</tr>
<tr>
<td>2292</td>
<td>Limit error, Out of reference memory</td>
</tr>
<tr>
<td>2293</td>
<td>Limit error, Selected channel is turned off</td>
</tr>
<tr>
<td>2301</td>
<td>Cursor error, Off-screen</td>
</tr>
<tr>
<td>2302</td>
<td>Cursor error, cursors in different frames</td>
</tr>
<tr>
<td>2311</td>
<td>Group requested has not been selected or has been deleted</td>
</tr>
</tbody>
</table>
Table 3–6 lists the device errors that can occur during digitizing oscilloscope operation. These errors may indicate that the oscilloscope needs repair.

Table 3–6: Device Error Messages – DDE Bit 3

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>Device-specific error</td>
</tr>
<tr>
<td>310</td>
<td>System error</td>
</tr>
<tr>
<td>311</td>
<td>Memory error</td>
</tr>
<tr>
<td>312</td>
<td>PUD memory lost</td>
</tr>
<tr>
<td>313</td>
<td>Calibration memory lost</td>
</tr>
<tr>
<td>314</td>
<td>Save/recall memory lost</td>
</tr>
<tr>
<td>315</td>
<td>Configuration memory lost</td>
</tr>
<tr>
<td>350</td>
<td>Queue overflow (does not set DDE bit)</td>
</tr>
</tbody>
</table>

Table 3–7 lists the system event messages. These messages are generated whenever certain system conditions occur.

Table 3–7: System Event Messages

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Query event</td>
</tr>
<tr>
<td>401</td>
<td>Power on (PON bit 7 set)</td>
</tr>
<tr>
<td>402</td>
<td>Operation complete (OPC bit 0 set)</td>
</tr>
<tr>
<td>403</td>
<td>User request (URQ bit 6 set)</td>
</tr>
<tr>
<td>404</td>
<td>Power fail (DDE bit 3 set)</td>
</tr>
<tr>
<td>405</td>
<td>Request control</td>
</tr>
<tr>
<td>410</td>
<td>Query INTERRUPTED (QYE bit 2 set)</td>
</tr>
<tr>
<td>420</td>
<td>Query UNTERMINATED (QYE bit 2 set)</td>
</tr>
<tr>
<td>430</td>
<td>Query DEADLOCKED (QYE bit 2 set)</td>
</tr>
<tr>
<td>440</td>
<td>Query UNTERMINATED after indefinite response (QYE bit 2 set)</td>
</tr>
<tr>
<td>450</td>
<td>Right menu button #1 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>451</td>
<td>Right menu button #2 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>452</td>
<td>Right menu button #3 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>453</td>
<td>Right menu button #4 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>454</td>
<td>Right menu button #5 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>460</td>
<td>Bottom menu button #1 pushed (URQ bit 6 set)</td>
</tr>
</tbody>
</table>
### Table 3–7: System Event Messages (Cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>461</td>
<td>Bottom menu button #2 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>462</td>
<td>Bottom menu button #3 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>463</td>
<td>Bottom menu button #4 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>464</td>
<td>Bottom menu button #5 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>465</td>
<td>Bottom menu button #6 pushed (URQ bit 6 set)</td>
</tr>
<tr>
<td>466</td>
<td>Bottom menu button #7 pushed (URQ bit 6 set)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Execution warning</td>
</tr>
<tr>
<td>510</td>
<td>String data too long, truncated</td>
</tr>
<tr>
<td>525</td>
<td>Parameter underrange</td>
</tr>
<tr>
<td>526</td>
<td>Parameter overrange</td>
</tr>
<tr>
<td>527</td>
<td>Parameter rounded</td>
</tr>
<tr>
<td>528</td>
<td>Parameter out of range</td>
</tr>
<tr>
<td>530</td>
<td>Data stop &gt; stop, Values swapped internally</td>
</tr>
<tr>
<td>531</td>
<td>Data stop &gt; record length, Curve truncated</td>
</tr>
<tr>
<td>532</td>
<td>Curve data too long, Curve truncated</td>
</tr>
<tr>
<td>540</td>
<td>Measurement warning</td>
</tr>
<tr>
<td>541</td>
<td>Measurement warning, Low signal amplitude</td>
</tr>
<tr>
<td>542</td>
<td>Measurement warning, Unstable histogram</td>
</tr>
<tr>
<td>543</td>
<td>Measurement warning, Low resolution</td>
</tr>
<tr>
<td>544</td>
<td>Measurement warning, Uncertain edge</td>
</tr>
<tr>
<td>545</td>
<td>Measurement warning, Invalid in minmax</td>
</tr>
<tr>
<td>546</td>
<td>Measurement warning, Need 3 edges</td>
</tr>
<tr>
<td>547</td>
<td>Measurement warning, Clipping positive/negative</td>
</tr>
<tr>
<td>548</td>
<td>Measurement warning, Clipping positive</td>
</tr>
<tr>
<td>549</td>
<td>Measurement warning, Clipping negative</td>
</tr>
<tr>
<td>550</td>
<td>InstaVu active – deactivate to see change</td>
</tr>
</tbody>
</table>
### Table 3–8: Execution Warning Messages – EXE Bit 4 (Cont.)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>551</td>
<td>InstaVu active – deactivate to use math</td>
</tr>
<tr>
<td>570</td>
<td>Saveref warning, decimated 500K waveform to 250K Ref (TDS 700A)</td>
</tr>
<tr>
<td>571</td>
<td>Option 05 not present</td>
</tr>
<tr>
<td>572</td>
<td>Option 2F not present</td>
</tr>
</tbody>
</table>

Table 3–9 shows internal errors that indicate an internal fault in the digitizing oscilloscope.

### Table 3–9: Internal Warning Messages

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>Internal warning</td>
</tr>
<tr>
<td>620</td>
<td>Internal warning, Bad thermistor</td>
</tr>
<tr>
<td>630</td>
<td>Internal warning, 50 Ω overload</td>
</tr>
</tbody>
</table>
Programming Examples

The example programs illustrate methods you can use to control the digitizing oscilloscope from the GPIB interface. The diskettes that come with this manual contain listings for these programs written in Microsoft QuickBASIC 4.5 and Microsoft QuickC 2.5.

The programs run on a PC-compatible system equipped with a Tektronix (National Instruments) GPIB board and associated drivers. For example, the programs will work with a Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB package (See Figure 4–1).

![Figure 4–1: Equipment Needed to Run the Example Programs](image)

All the example programs assume that the GPIB system recognizes the digitizing oscilloscope as DEV1 and the PC (controller) as GPIB0. You can use the IBCONF.EXE program to assign these names.

The example software includes:

- **MEAS**: automatically measures waveform parameters.
- **COMM**: shows communication between controller and oscilloscope.
Programming Examples

- GETWFM: reads a waveform from an oscilloscope and stores it in a file.
- CURSOR: uses cursors to measure waveform parameters.
- TL: a talker-listener program.

Compiling the Example Programs

The example programs diskette contains programs written in Microsoft QuickBASIC 4.5 and Microsoft QuickC 2.5.

Executable versions of the programs are in the PROGRAMS directory. Source versions are in the SOURCES directory. Within this directory, the QuickBASIC programs are in the Q-BASIC subdirectory and the QuickC programs are in the QUICK-C subdirectory.

A README file in each directory explains how to build executable code from the source files provided.

The QuickC directory also comes with sample MAKE files and sample executable files. These have the suffix .MAK.

If you wish to develop code, you will need to use files that come with the GPIB system. Specifically, the QuickBASIC programs use QBDECL.BAS and QB1B.OBJ. The QuickC programs use DECL.H and MC1B.OBJ.

**NOTE** The programs you compile in the Sources directory work with the Tektronix S3FG210 (National Instruments GPIB-PCII-IIA) GPIB system. It may take extra steps or changes to get them to work with older Tektronix GURU and other GPIB systems.

Compiling and Linking Your Example Quick-C Programs

To make an executable for any example, perform the following:

1. Install QuickC. Select the SMALL memory model. Be sure to set up your path so DOS can access the QuickC directory.

2. Install the Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB board and drivers. Remember to identify the GPIB device as DEV1. You can use the IBCONF.EXE program to do this.

3. Copy the files from the examples diskette to your hard disk. You might also create a special directory to store them. For example, if the current drive is hard disk C, you want to store the examples in drive C, and the examples diskette is in drive B, you might type:

   ```
   mkdir examples
   ```
cd examples

4. For this installation, you will also want to copy DECL.H and MCIB.OBJ from your Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB drivers directory to this directory. For example, if the GPIB drivers are in the gpi-b-pc directory and you are in the example programs directory, you would type:

   copy \gpi-b-pc\decl.h .
   copy \gpi-b-pc\mcib.obj .

5. To compile and link your TDS sample C programs, simply type: nmake <file name>.mak

   where <file name> refers to the name of the example program you wish to compile and link. Specifically:

   To compile and link MEAS.C, type: nmake meas.mak
   To compile and link COMM.C, type: nmake comm.mak
   To compile and link GETWF.M.C, type: nmake getwf.mak
   To compile and link CURSOR.C, type: nmake cursor.mak
   To compile and link TL.C, type: nmake tl.mak

6. Run the program by simply typing the program name.

   To run meas, type: meas
   To run comm, type: comm
   To run getwf, type: getwf
   To run cursor, type: cursor
   To run tl, type: tl

To make an executable for any of the following files, perform the following:

1. Install QuickBASIC.

2. Install the Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB board and drivers. Remember to reboot your PC to initialize the GPIB drivers.

3. Copy the files from the examples diskette to your hard disk. You might also create a special directory to store them. For example, if the current drive is
hard disk C, you want to store the examples in drive C, and the examples diskette is in drive B, you might type:

mkdir examples

cd examples

copy b:\q-basic\*. *

4. For this installation, you will also want to copy QBDECL.BAS and QBIB.OBJ from your Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB drivers directory to the directory your example programs are in. For example, if the GPIB drivers are in the gpib-pc directory and you are in the example programs directory, you would type:

copy \gpib-pc\qbdecl.bas .
copy \gpib-pc\qbib.obj .

5. Perform the following two steps for example programs:

a. Compile the program by using the following command:

   bc /o <file>.bas;

   where <file> is one of the example program names.

   To compile MEAS.BAS, type: bc /o meas.bas;
   To compile COMM.BAS, type: bc /o comm.bas;
   To compile GETWFM.BAS, type: bc /o getwfm.bas;
   To compile CURSOR.BAS, type: bc /o cursor.bas;
   To compile TL.BAS, type: bc /o t1.bas;

b. Link the compiled program with the qbib.obj module to create the executable program (file.EXE) by using the following command:

   link <file>.obj+qbib.obj;

   where <file> is one of the above program names.

   To link MEAS.OBJ, type: link meas.obj+qbib.obj;
   To link COMM.OBJ, type: link comm.obj+qbib.obj;
   To link GETWFM.OBJ, type: link getwfm.obj+qbib.obj;
   To link CURSOR.OBJ, type: link cursor.obj+qbib.obj;
   To link TL.OBJ, type: link t1.obj+qbib.obj;
GPIBIO.BAS is a collection of input/output routines used by the other programs and is included for proper file compilation.

6. Run the program by simply typing the program name.
   
   To run meas, type: meas
   To run comm, type: comm
   To run getwfm, type: getwfm
   To run cursor, type: cursor
   To run t1, type: t1

**NOTE.** The example programs disable front-panel operation while they are running and reenable it when they terminate. If your program terminates prematurely, front-panel operation may remain disabled. To reenable front-panel operation, do one of the following: cycle power on the digitizing oscilloscope or send the GPIB command UNLOCK ALL to unlock the front panel. You can send the UNLOCK ALL command with the TL program included in your sample programs disk.
Appendix A: Character Charts

The characters in Table A–1 are available for the digitizing oscilloscope. Numbers in the lower left corners are character widths in pixels.

Table A–1: The TDS Character Set

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>\</td>
<td>space</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>i</td>
</tr>
<tr>
<td>1</td>
<td>2511</td>
<td>Ω</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>ç</td>
<td>Δ</td>
<td>”</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>Ç</td>
<td>BW</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>..</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>`</td>
<td>µ</td>
<td>&amp;</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
</tr>
<tr>
<td>6</td>
<td>ß</td>
<td>µ</td>
<td>©</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
</tr>
<tr>
<td>7</td>
<td>'</td>
<td>~</td>
<td>(</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
</tr>
<tr>
<td>8</td>
<td>i</td>
<td>～</td>
<td>)</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
</tr>
<tr>
<td>9</td>
<td>HT</td>
<td>—</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
</tr>
<tr>
<td>A</td>
<td>LF</td>
<td>∞</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
</tr>
<tr>
<td>B</td>
<td>`</td>
<td>ESC</td>
<td>+</td>
<td>;</td>
<td>K</td>
<td>[</td>
<td>k</td>
</tr>
<tr>
<td>C</td>
<td>±</td>
<td>x</td>
<td>,</td>
<td>,</td>
<td>L</td>
<td>\</td>
<td>l</td>
</tr>
<tr>
<td>D</td>
<td>CR</td>
<td>≠</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td>]</td>
</tr>
<tr>
<td>E</td>
<td>–</td>
<td>~</td>
<td>.</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
</tr>
<tr>
<td>F</td>
<td>•</td>
<td>•</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>–</td>
<td>o</td>
</tr>
</tbody>
</table>
## Appendix A: Character Charts

### Table A-2: ASCII & GPIB Code Chart

<table>
<thead>
<tr>
<th>B7</th>
<th>B6</th>
<th>B5</th>
<th>B4</th>
<th>B3</th>
<th>B2</th>
<th>B1</th>
<th>CONTROL</th>
<th>NUMBERS SYMBOLS</th>
<th>UPPER CASE</th>
<th>LOWER CASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>@</td>
<td>A0</td>
<td>P</td>
<td>SA0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>40</td>
<td>60</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>16</td>
<td>20</td>
<td>32</td>
<td>48</td>
<td>20</td>
<td>32</td>
<td>48</td>
<td>101</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>41</td>
<td>61</td>
<td>17</td>
<td>101</td>
<td>TA1</td>
<td>TA17</td>
<td>141</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>21</td>
<td>33</td>
<td>49</td>
<td>41</td>
<td>A</td>
<td>Q</td>
<td>a</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>22</td>
<td>18</td>
<td>42</td>
<td>62</td>
<td>18</td>
<td>102</td>
<td>TA2</td>
<td>TA18</td>
<td>142</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>12</td>
<td>18</td>
<td>22</td>
<td>34</td>
<td>50</td>
<td>42</td>
<td>B</td>
<td>R</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>19</td>
<td>23</td>
<td>15</td>
<td>27</td>
<td>53</td>
<td>45</td>
<td>LA5</td>
<td>da</td>
<td>c</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>19</td>
<td>23</td>
<td>23</td>
<td>35</td>
<td>51</td>
<td>43</td>
<td>#</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>20</td>
<td>24</td>
<td>14</td>
<td>26</td>
<td>52</td>
<td>44</td>
<td>T</td>
<td>T</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>25</td>
<td>31</td>
<td>21</td>
<td>37</td>
<td>53</td>
<td>54</td>
<td>LA5</td>
<td>da</td>
<td>c</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>25</td>
<td>31</td>
<td>23</td>
<td>35</td>
<td>51</td>
<td>43</td>
<td>#</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>@</td>
<td>A0</td>
<td>P</td>
<td>SA0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16</td>
<td>20</td>
<td>32</td>
<td>48</td>
<td>20</td>
<td>32</td>
<td>48</td>
<td>101</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>16</td>
<td>41</td>
<td>61</td>
<td>17</td>
<td>101</td>
<td>TA1</td>
<td>TA17</td>
<td>141</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>21</td>
<td>33</td>
<td>49</td>
<td>41</td>
<td>A</td>
<td>Q</td>
<td>a</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>22</td>
<td>18</td>
<td>42</td>
<td>62</td>
<td>18</td>
<td>102</td>
<td>TA2</td>
<td>TA18</td>
<td>142</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>12</td>
<td>18</td>
<td>22</td>
<td>34</td>
<td>50</td>
<td>42</td>
<td>B</td>
<td>R</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>19</td>
<td>23</td>
<td>15</td>
<td>27</td>
<td>53</td>
<td>45</td>
<td>LA5</td>
<td>da</td>
<td>c</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>19</td>
<td>23</td>
<td>23</td>
<td>35</td>
<td>51</td>
<td>43</td>
<td>#</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>20</td>
<td>24</td>
<td>14</td>
<td>26</td>
<td>52</td>
<td>44</td>
<td>T</td>
<td>T</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>25</td>
<td>31</td>
<td>21</td>
<td>37</td>
<td>53</td>
<td>54</td>
<td>LA5</td>
<td>da</td>
<td>c</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>25</td>
<td>31</td>
<td>23</td>
<td>35</td>
<td>51</td>
<td>43</td>
<td>#</td>
<td>!</td>
<td></td>
</tr>
</tbody>
</table>

### KEY
- Octal: 5
- ENQ: 5
- GPIB code (with ATN asserted): 5
- ASCII character: 5
- Decimal: 5

---

TDS Family Oscilloscope Programmer Manual
Appendix B: Reserved Words

The following is a list of the reserved words of the digitizing oscilloscope. Do not use these words for aliases. Capital letters identify the required minimum spelling. Hint: Use the full spelling for the most robust code as the minimum spelling rules may change over time and from model to model.

<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Reserved Word</th>
<th>Reserved Word</th>
<th>Reserved Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CAL</td>
<td>BACKGround</td>
<td>CLEARSpool</td>
<td>DISplay</td>
</tr>
<tr>
<td>*CLS</td>
<td>BACKWards</td>
<td>CLOCk</td>
<td>DOTs</td>
</tr>
<tr>
<td>*DDT</td>
<td>BANwidth</td>
<td>CMEan</td>
<td>DPU411</td>
</tr>
<tr>
<td>*ESE</td>
<td>BASE</td>
<td>COLLision</td>
<td>DPU412</td>
</tr>
<tr>
<td>*ESR</td>
<td>BAUd</td>
<td>COLOR</td>
<td>DUAI</td>
</tr>
<tr>
<td>*IDN</td>
<td>BELJ</td>
<td>COMPARE</td>
<td>ECL</td>
</tr>
<tr>
<td>*LRN</td>
<td>BIN</td>
<td>CONTRast</td>
<td>EDGE</td>
</tr>
<tr>
<td>*OPC</td>
<td>BIT_Nr</td>
<td>CONTROL</td>
<td>EDGE1</td>
</tr>
<tr>
<td>*PSC</td>
<td>BLACKman</td>
<td>COPY</td>
<td>EDGE2</td>
</tr>
<tr>
<td>*PUD</td>
<td>BMP</td>
<td>COUNT</td>
<td>Either</td>
</tr>
<tr>
<td>*RCL</td>
<td>BMPCOLOR</td>
<td>COUPling</td>
<td>ENCdg</td>
</tr>
<tr>
<td>*RST</td>
<td>BN_Fmt</td>
<td>CPU</td>
<td>ENV</td>
</tr>
<tr>
<td>*SAV</td>
<td>BOLD</td>
<td>CRMs</td>
<td>ENVelope</td>
</tr>
<tr>
<td>*SRE</td>
<td>BOTH</td>
<td>CROSSHhair</td>
<td>EPSColor</td>
</tr>
<tr>
<td>*STB</td>
<td>BOTTOM1</td>
<td>CRVchk</td>
<td>EPSColorImg</td>
</tr>
<tr>
<td>*TRG</td>
<td>BOTTOM2</td>
<td>CURRENT</td>
<td>EPSImage</td>
</tr>
<tr>
<td>*TST</td>
<td>BOTTOM3</td>
<td>CURSor</td>
<td>EPSMono</td>
</tr>
<tr>
<td>*WAI</td>
<td>BOTTOM4</td>
<td>CURSOR1</td>
<td>EPSOn</td>
</tr>
<tr>
<td>ABOrt</td>
<td>BOTTOM5</td>
<td>CURSOR2</td>
<td>EVEN</td>
</tr>
<tr>
<td>ABSolute</td>
<td>BOTTOM6</td>
<td>CURVe</td>
<td>EVENT</td>
</tr>
<tr>
<td>AC</td>
<td>BOTTOM7</td>
<td>CUSTom</td>
<td>EVENT</td>
</tr>
<tr>
<td>ACCEpt</td>
<td>BOX</td>
<td>CWD</td>
<td>EVENT</td>
</tr>
<tr>
<td>ACQuire</td>
<td>BURst</td>
<td>DATa</td>
<td>EVENTSTime</td>
</tr>
<tr>
<td>ACQUISition</td>
<td>BUSY</td>
<td>DATE</td>
<td>EVMsg</td>
</tr>
<tr>
<td>ACTivate</td>
<td>BY</td>
<td>DC</td>
<td>EVQty</td>
</tr>
<tr>
<td>ACTUal</td>
<td>BYCONtents</td>
<td>DEFault</td>
<td>EXECute</td>
</tr>
<tr>
<td>ALLas</td>
<td>BYT_Nr</td>
<td>DEFINE</td>
<td>FACTory</td>
</tr>
<tr>
<td>ALL</td>
<td>BYT_Or</td>
<td>DELay</td>
<td>FALL</td>
</tr>
<tr>
<td>ALLev</td>
<td>CALibrate</td>
<td>DELAYEd</td>
<td>FALSE</td>
</tr>
<tr>
<td>ALLocate</td>
<td>CATalog</td>
<td>DELete</td>
<td>FASTERthan</td>
</tr>
<tr>
<td>AMPlitude</td>
<td>CARea</td>
<td>DELTa</td>
<td>FFT</td>
</tr>
<tr>
<td>AND</td>
<td>CENtronicss</td>
<td>DELTAtime</td>
<td>FIELD</td>
</tr>
<tr>
<td>APPMenu</td>
<td>CH1</td>
<td>DELWarn</td>
<td>FIELD1</td>
</tr>
<tr>
<td>AREa</td>
<td>CH2</td>
<td>DESE</td>
<td>FIELD2</td>
</tr>
<tr>
<td>ARMed</td>
<td>CH3</td>
<td>DESKJET</td>
<td>FIELDEither</td>
</tr>
<tr>
<td>ASC</td>
<td>CH4</td>
<td>DESKJETC</td>
<td>FIELDRATE</td>
</tr>
<tr>
<td>ASCli</td>
<td>CHKsm0</td>
<td>DESTination</td>
<td>FIELDS</td>
</tr>
<tr>
<td>AUTO</td>
<td>CLASS</td>
<td>DIAg</td>
<td>FIELDType</td>
</tr>
<tr>
<td>AUTOSet</td>
<td>CLEAR</td>
<td>DIR</td>
<td>FiFty</td>
</tr>
<tr>
<td>AUXiliary</td>
<td>CLEARMenu</td>
<td>DIRECTION</td>
<td></td>
</tr>
<tr>
<td>AVERAGE</td>
<td>CLEARSNapshot</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix B: Reserved Words

<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILEName</td>
<td>INTENSI!Fied</td>
</tr>
<tr>
<td>FILESystem</td>
<td>NDUty</td>
</tr>
<tr>
<td>FILTER</td>
<td>NEGA!ve</td>
</tr>
<tr>
<td>FIRST</td>
<td>NGSynchwidth</td>
</tr>
<tr>
<td>FITtoscreen</td>
<td>NEWpass</td>
</tr>
<tr>
<td>FLAG</td>
<td>NOISRej</td>
</tr>
<tr>
<td>FLEXformat</td>
<td>NONe</td>
</tr>
<tr>
<td>FORCE</td>
<td>NOR</td>
</tr>
<tr>
<td>FORMAT</td>
<td>NORMal</td>
</tr>
<tr>
<td>FORWards</td>
<td>NOVershoot</td>
</tr>
<tr>
<td>FPAnel</td>
<td>NR_Pt</td>
</tr>
<tr>
<td>FRAME</td>
<td>NTSc</td>
</tr>
<tr>
<td>FREE</td>
<td>NUMACq</td>
</tr>
<tr>
<td>FREESpace</td>
<td>NUMAVg</td>
</tr>
<tr>
<td>FREQUENCY</td>
<td>NUMEnv</td>
</tr>
<tr>
<td>FULL</td>
<td>NUMERic</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>NWidth</td>
</tr>
<tr>
<td>GATING</td>
<td>ODD</td>
</tr>
<tr>
<td>GLitch</td>
<td>OFF</td>
</tr>
<tr>
<td>GND</td>
<td>OFFSet</td>
</tr>
<tr>
<td>GPIb</td>
<td>ON</td>
</tr>
<tr>
<td>GRAticule</td>
<td>ONCE</td>
</tr>
<tr>
<td>GRAYscale</td>
<td>OR</td>
</tr>
<tr>
<td>GRID</td>
<td>OPTION</td>
</tr>
<tr>
<td>HALT</td>
<td>OUTside</td>
</tr>
<tr>
<td>HAMmung</td>
<td>OVERAll</td>
</tr>
<tr>
<td>HARDCopy</td>
<td>OVERWrite</td>
</tr>
<tr>
<td>HARDFlagging</td>
<td>PA!red</td>
</tr>
<tr>
<td>HBArs</td>
<td>PAL</td>
</tr>
<tr>
<td>HDELTA</td>
<td>PALEtte</td>
</tr>
<tr>
<td>HDR</td>
<td>PARity</td>
</tr>
<tr>
<td>HDTV</td>
<td>PARTial</td>
</tr>
<tr>
<td>HEADER</td>
<td>PASSWord</td>
</tr>
<tr>
<td>HERTz</td>
<td>P!ttern</td>
</tr>
<tr>
<td>HFRej</td>
<td>PCX</td>
</tr>
<tr>
<td>HIGH</td>
<td>PC!COLOR</td>
</tr>
<tr>
<td>HIGHLimit</td>
<td>PDUty</td>
</tr>
<tr>
<td>HIRes</td>
<td>PEAKdetect</td>
</tr>
<tr>
<td>HISTogram</td>
<td>PERCent</td>
</tr>
<tr>
<td>HOLDOff</td>
<td>PERIod</td>
</tr>
<tr>
<td>HOLDTime</td>
<td>PERSistence</td>
</tr>
<tr>
<td>HORIZontal</td>
<td>PH!se</td>
</tr>
<tr>
<td>HPGL</td>
<td>PK2pk</td>
</tr>
<tr>
<td>HPOS1</td>
<td>POLarity</td>
</tr>
<tr>
<td>HPOS2</td>
<td>PORT</td>
</tr>
<tr>
<td>HUNDred</td>
<td>PORTR!t</td>
</tr>
<tr>
<td>ID</td>
<td>Position</td>
</tr>
<tr>
<td>IMMed</td>
<td>POSITION1</td>
</tr>
<tr>
<td>IMPedance</td>
<td>POSITION2</td>
</tr>
<tr>
<td>INDependent</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>INIT</td>
<td>POVershoot</td>
</tr>
<tr>
<td>INFPersit</td>
<td>PREView</td>
</tr>
<tr>
<td>INIT</td>
<td>PRInt</td>
</tr>
<tr>
<td>INPUT</td>
<td>PROBE</td>
</tr>
<tr>
<td>MG tones</td>
<td>PT_Fmt</td>
</tr>
<tr>
<td>OFF</td>
<td>PT_Off</td>
</tr>
<tr>
<td>PULSE</td>
<td>PW!dth</td>
</tr>
<tr>
<td>RATE</td>
<td>RATE1</td>
</tr>
<tr>
<td>RATE1</td>
<td>RATE2</td>
</tr>
<tr>
<td>RATE2</td>
<td>RATE3</td>
</tr>
<tr>
<td>RATE3</td>
<td>RATE4</td>
</tr>
<tr>
<td>REAdy</td>
<td>RECall</td>
</tr>
<tr>
<td>RECALL</td>
<td>REColorlength</td>
</tr>
<tr>
<td>RECTangular</td>
<td>REF</td>
</tr>
<tr>
<td>REF1</td>
<td>REF2</td>
</tr>
<tr>
<td>REF2</td>
<td>REF3</td>
</tr>
<tr>
<td>REF3</td>
<td>REF4</td>
</tr>
<tr>
<td>REF4</td>
<td>REFl!vel</td>
</tr>
<tr>
<td>REGular</td>
<td>REGular</td>
</tr>
<tr>
<td>REM</td>
<td>REJect</td>
</tr>
<tr>
<td>RENAME</td>
<td>REM</td>
</tr>
<tr>
<td>REPEt</td>
<td>RESET</td>
</tr>
<tr>
<td>RESET</td>
<td>RESETAll</td>
</tr>
<tr>
<td>RESETAll</td>
<td>RESULT</td>
</tr>
<tr>
<td>RIS!e</td>
<td>RI</td>
</tr>
<tr>
<td>RIBinary</td>
<td>RIL! binary</td>
</tr>
<tr>
<td>RIGHT1</td>
<td>RIGHT2</td>
</tr>
<tr>
<td>RIGHT2</td>
<td>RIGHT3</td>
</tr>
<tr>
<td>RIGHT3</td>
<td>RIGHT4</td>
</tr>
<tr>
<td>RIGHT4</td>
<td>RIGHT5</td>
</tr>
<tr>
<td>RIGHT5</td>
<td>RIS!e</td>
</tr>
<tr>
<td>RLE</td>
<td>R!e</td>
</tr>
<tr>
<td>RMD!r</td>
<td>RMS</td>
</tr>
<tr>
<td>ROLL</td>
<td>R!l</td>
</tr>
<tr>
<td>RP</td>
<td>RP! Binary</td>
</tr>
<tr>
<td>RPB!inary</td>
<td>RS232</td>
</tr>
<tr>
<td>RUN</td>
<td>RUNSAfter</td>
</tr>
<tr>
<td>RUNSAfter</td>
<td>RUNSTop</td>
</tr>
<tr>
<td>RUNSTop</td>
<td>RUNT</td>
</tr>
<tr>
<td>R!NT</td>
<td>SAMple</td>
</tr>
<tr>
<td>SAMple</td>
<td>SAVe</td>
</tr>
<tr>
<td>SCA!e</td>
<td>SCAN</td>
</tr>
<tr>
<td>SCAN</td>
<td>SCANPeriod</td>
</tr>
<tr>
<td>SCROLLBAR</td>
<td>SCROLLTEXT</td>
</tr>
<tr>
<td>SCROLLTEXT</td>
<td></td>
</tr>
<tr>
<td>SECAm</td>
<td>STANdard</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>SECdiv</td>
<td>START</td>
</tr>
<tr>
<td>SECOnds</td>
<td>STATE</td>
</tr>
<tr>
<td>SELect</td>
<td>STOP</td>
</tr>
<tr>
<td>SEQUENCE</td>
<td>STOPAfter</td>
</tr>
<tr>
<td>SET</td>
<td>STOPBits</td>
</tr>
<tr>
<td>SETHold</td>
<td>STORE</td>
</tr>
<tr>
<td>SETLevel</td>
<td>STYLE</td>
</tr>
<tr>
<td>SETTime</td>
<td>SYNC</td>
</tr>
<tr>
<td>SETUp</td>
<td>SYStem</td>
</tr>
<tr>
<td>SHORT</td>
<td>TARget</td>
</tr>
<tr>
<td>SHOW</td>
<td>TEKSecure</td>
</tr>
<tr>
<td>SINX</td>
<td>TEMPERature</td>
</tr>
<tr>
<td>SLEWRate</td>
<td>TEMPLate</td>
</tr>
<tr>
<td>SLOpe</td>
<td>TEXT</td>
</tr>
<tr>
<td>SLOWERthan</td>
<td>THLinkjet</td>
</tr>
<tr>
<td>SNAP</td>
<td>THReshold</td>
</tr>
<tr>
<td>SNAPShot</td>
<td>TIFF</td>
</tr>
<tr>
<td>SOFTFlagging</td>
<td>TIMe</td>
</tr>
<tr>
<td>SOURCE</td>
<td>TIMEOut</td>
</tr>
<tr>
<td>SOURCE1</td>
<td>TITLE</td>
</tr>
<tr>
<td>SOURCE2</td>
<td>TO</td>
</tr>
<tr>
<td>SPECTral</td>
<td>TOlerance</td>
</tr>
<tr>
<td>SRIbinary</td>
<td>TRACk</td>
</tr>
<tr>
<td>SRPbinary</td>
<td>TRIGAfter</td>
</tr>
</tbody>
</table>
Appendix B: Reserved Words
Appendix C: Interface Specifications

This appendix describes details of the GPIB remote interface of the digitizing oscilloscope. Normally, you will not need this information to use the digitizing oscilloscope, but the information is useful when connecting to controllers of unusual configuration.

GPIB Function Subsets

The digitizing oscilloscope supports many GPIB function subsets, as listed below. Some of the listings describe subsets that the digitizing oscilloscope does not support.

- **SH1 (Source Handshake)**. The digitizing oscilloscope can transmit multiline messages across the GPIB.
- **AH1 (Acceptor Handshake)**. The digitizing oscilloscope can receive multiline messages across the GPIB.
- **T5 (Talker)**. The digitizing oscilloscope becomes a talker when its talk address is sent with the ATN (Attention) line asserted. It can send both response data and status information when responding to a serial poll. It ceases to be a talker when the talk address if another device is sent with ATN asserted. The digitizing oscilloscope has talk-only capability for hardcopy operation.
- **L4 (Listener)**. The digitizing oscilloscope becomes a listener when its listen address is sent with the ATN (Attention) line asserted. The digitizing oscilloscope does not have listen-only capability.
- **SR1 (Service Request)**. The digitizing oscilloscope asserts an SRQ (Service Request) line to notify the controller when it requires service.
- **RL1 (Remote/Local)**. The digitizing oscilloscope responds to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.
- **PP0 (Parallel Poll)**. The digitizing oscilloscope has no parallel poll capability. It does not respond to the following interface messages: PPC, PPD, PPE, and PPU. The digitizing oscilloscope does not send out a status message when the ATN (Attention) and EOI (End or Identify) lines are asserted simultaneously.
- **DC1 (Device Clear)**. The digitizing oscilloscope responds to the DCL (Device Clear) and, when made a listener, the SDC (Selected Device Clear) interface messages.
DT1 (Device Trigger). When acting as a listener, the digitizing oscilloscope responds to the GET (Group Execute Trigger) interface message.

C0 (Controller). The digitizing oscilloscope cannot control other devices.

E2 (Electrical). The digitizing oscilloscope uses tristate buffers to provide optimal high-speed data transfer.

**Interface Messages**

Table C–1 shows the standard interface messages that are supported by the digitizing oscilloscope.

**Table C–1: TDS Family Oscilloscope Standard Interface Message**

<table>
<thead>
<tr>
<th>Message</th>
<th>GPIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCL</td>
<td>Yes</td>
</tr>
<tr>
<td>GET</td>
<td>Yes</td>
</tr>
<tr>
<td>GTL</td>
<td>Yes</td>
</tr>
<tr>
<td>LLO</td>
<td>Yes</td>
</tr>
<tr>
<td>PPC</td>
<td>No</td>
</tr>
<tr>
<td>PPD</td>
<td>No</td>
</tr>
<tr>
<td>PPE</td>
<td>No</td>
</tr>
<tr>
<td>PPU</td>
<td>No</td>
</tr>
<tr>
<td>SDC</td>
<td>Yes</td>
</tr>
<tr>
<td>SPD</td>
<td>Yes</td>
</tr>
<tr>
<td>SPE</td>
<td>Yes</td>
</tr>
<tr>
<td>TCT</td>
<td>No</td>
</tr>
<tr>
<td>UNL</td>
<td>Yes</td>
</tr>
<tr>
<td>UNT</td>
<td>Yes</td>
</tr>
<tr>
<td>Listen Addresses</td>
<td>Yes</td>
</tr>
<tr>
<td>Talk Addresses</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The factory initialization settings provide you a known state for the digitizing oscilloscope.

Factory initialization sets values as shown in Table D–1.

### Table D–1: Factory Initialization Settings

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquire mode</td>
<td>Sample</td>
</tr>
<tr>
<td>Acquire repetitive signal</td>
<td>TDS 400A, 510A, 500B, &amp; 700A: ON (Enable ET)</td>
</tr>
<tr>
<td>Acquire stop after</td>
<td>RUN/STOP button only</td>
</tr>
<tr>
<td>Acquire # of averages</td>
<td>16</td>
</tr>
<tr>
<td>Acquire # of envelopes</td>
<td>10</td>
</tr>
<tr>
<td>Channel selection</td>
<td>Channel 1 on, all others off</td>
</tr>
<tr>
<td>Cursor H Bar 1 position</td>
<td>10% of graticule height (~3.2 divs from the center)</td>
</tr>
<tr>
<td>Cursor H Bar 2 position</td>
<td>90% of the graticule height (+3.2 divs from the center)</td>
</tr>
<tr>
<td>Cursor V Bar 1 position</td>
<td>10% of the record length</td>
</tr>
<tr>
<td>Cursor V Bar 2 position</td>
<td>90% of the record length</td>
</tr>
<tr>
<td>Cursor amplitude units</td>
<td>Base</td>
</tr>
<tr>
<td>Cursor function</td>
<td>Off</td>
</tr>
<tr>
<td>Cursor mode</td>
<td>Independent</td>
</tr>
<tr>
<td>Cursor time units</td>
<td>Seconds</td>
</tr>
<tr>
<td>Date and time</td>
<td>No change</td>
</tr>
<tr>
<td>Delay events, triggerable after main</td>
<td>TDS 400A: 1</td>
</tr>
<tr>
<td></td>
<td>TDS 510A, 500B, 600B, &amp; 700A: 2</td>
</tr>
<tr>
<td>Delay time, delayed runs after main</td>
<td>TDS 400A: 10 ns</td>
</tr>
<tr>
<td></td>
<td>TDS 510A, 500B, 600B, &amp; 700A: 16 ns</td>
</tr>
<tr>
<td>Delay time, delayed triggerable after main</td>
<td>TDS 400A: 60 ns</td>
</tr>
<tr>
<td></td>
<td>TDS 510A, 500B, 600B, &amp; 700A: 16 ns</td>
</tr>
<tr>
<td>Delay trigger average #</td>
<td>16</td>
</tr>
<tr>
<td>Delay trigger envelope #</td>
<td>10</td>
</tr>
<tr>
<td>Delayed, delay by ...</td>
<td>Delay by Time</td>
</tr>
</tbody>
</table>
### Table D–1: Factory Initialization Settings (Cont.)

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delayed edge trigger coupling (TDS 400A only)</td>
<td>DC Main Trigger</td>
</tr>
<tr>
<td></td>
<td>0 V</td>
</tr>
<tr>
<td>Delayed edge trigger slope</td>
<td>Rising</td>
</tr>
<tr>
<td>Delayed edge trigger source</td>
<td>Channel 1</td>
</tr>
<tr>
<td>Delayed, time base mode</td>
<td>Delayed Runs After Main</td>
</tr>
<tr>
<td>Deskew, Channel/Probe</td>
<td>0 seconds</td>
</tr>
<tr>
<td>Display clock</td>
<td>No Change</td>
</tr>
<tr>
<td>Display color – collision contrast</td>
<td>TDS 644B, 684B, &amp; 700A: Off</td>
</tr>
<tr>
<td>Display color – map math colors</td>
<td>TDS 644B, 684B, &amp; 700A: Color ‘Math’</td>
</tr>
<tr>
<td>Display color – map reference colors</td>
<td>TDS 644B, 684B, &amp; 700A: Color ‘Ref’</td>
</tr>
<tr>
<td>Display color – palette</td>
<td>TDS 644B, 684B, &amp; 700A: Normal</td>
</tr>
<tr>
<td>Display color – palette colors</td>
<td>TDS 644B, 684B, &amp; 700A: The colors of each palette are reset to factory hue, saturation, and lightness (HLS) values</td>
</tr>
<tr>
<td>Display color – persistence palette</td>
<td>TDS 644B, 684B, &amp; 700A: Temperature</td>
</tr>
<tr>
<td>Display format</td>
<td>YT</td>
</tr>
<tr>
<td>Display graticule type</td>
<td>Full</td>
</tr>
<tr>
<td>Display instavu persistence</td>
<td>TDS 500B &amp; 700A: Varpersist</td>
</tr>
<tr>
<td>Display instavu style</td>
<td>TDS 500B &amp; 700A: Vectors</td>
</tr>
<tr>
<td>Display instavu varpersist</td>
<td>TDS 500B &amp; 700A: 500 e–3</td>
</tr>
<tr>
<td>Display intensity – contrast</td>
<td>TDS 400A, 500B, 620B, &amp; 680B: 175% TDS 510A: 150%</td>
</tr>
<tr>
<td>Display interpolation filter</td>
<td>Sin(x)/x</td>
</tr>
<tr>
<td>Display mode</td>
<td>Normal</td>
</tr>
<tr>
<td>Display style</td>
<td>Vectors</td>
</tr>
</tbody>
</table>
### Table D–1: Factory Initialization Settings (Cont.)

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display trigger bar style</td>
<td>Short</td>
</tr>
<tr>
<td>Display trigger “T”</td>
<td>On</td>
</tr>
<tr>
<td>Display variable persistence</td>
<td>500 ms</td>
</tr>
<tr>
<td>Edge trigger coupling</td>
<td>DC</td>
</tr>
<tr>
<td>Edge trigger level</td>
<td>0.0 V</td>
</tr>
<tr>
<td>Edge trigger slope</td>
<td>Rising</td>
</tr>
<tr>
<td>Edge trigger source</td>
<td>Channel 1</td>
</tr>
<tr>
<td>GPIB parameters</td>
<td>No change</td>
</tr>
<tr>
<td>Hardcopy Format</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Layout</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Palette</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Port</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Horizontal – delay time/division</td>
<td>50 μs</td>
</tr>
<tr>
<td>Horizontal – delay trigger position</td>
<td>50%</td>
</tr>
<tr>
<td>Horizontal – delay trigger record length</td>
<td>500 points (10 divs)</td>
</tr>
<tr>
<td>Horizontal – FastFrame</td>
<td>TDS 500B &amp; 700A: Off</td>
</tr>
<tr>
<td>Horizontal – FastFrame, frame count</td>
<td>TDS 500B &amp; 700A: 2</td>
</tr>
<tr>
<td>Horizontal – FastFrame, frame length</td>
<td>500</td>
</tr>
<tr>
<td>Horizontal – fit to screen</td>
<td>Off</td>
</tr>
<tr>
<td>Horizontal – main time/division</td>
<td>500 μs</td>
</tr>
<tr>
<td>Horizontal – main trigger position</td>
<td>50%</td>
</tr>
<tr>
<td>Horizontal – main trigger record length</td>
<td>500 points (10 divs)</td>
</tr>
<tr>
<td>Horizontal – time base</td>
<td>Main only</td>
</tr>
<tr>
<td>Limit template</td>
<td>±V Limit 40 mdiv</td>
</tr>
<tr>
<td></td>
<td>±H Limit 40 mdiv</td>
</tr>
<tr>
<td>Limit template destination</td>
<td>Ref1</td>
</tr>
<tr>
<td>Limit template source</td>
<td>Ch1</td>
</tr>
<tr>
<td>Limit test sources</td>
<td>Ch1 compared to Ref1; all others compared to none.</td>
</tr>
<tr>
<td>Limit testing</td>
<td>Off</td>
</tr>
<tr>
<td>Limit testing – hardcopy if condition met</td>
<td>Off</td>
</tr>
<tr>
<td>Limit testing – ring bell if condition met</td>
<td>Off</td>
</tr>
<tr>
<td>Logic pattern trigger Ch4 (Ax2) input</td>
<td>TDS 510A, 500B, 600B, &amp; 700A: X (do not care)</td>
</tr>
<tr>
<td>Logic state trigger Ch4 (Ax2) input</td>
<td>TDS 510A, 500B, 600B, &amp; 700A: Rising edge</td>
</tr>
</tbody>
</table>
## Table D–1: Factory Initialization Settings (Cont.)

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic trigger class</td>
<td>TDS 500B, 600B, &amp; 700A: Pattern</td>
</tr>
<tr>
<td>Logic trigger input (pattern and state)</td>
<td>TDS 510A, 500B, 600B, &amp; 700A: Channel 1 = H (high),</td>
</tr>
<tr>
<td></td>
<td>Channels 2 &amp; 3 (Ax1) = X (do not care)</td>
</tr>
<tr>
<td>Logic trigger logic (pattern and state)</td>
<td>TDS 510A, 500B, 600B, &amp; 700A: AND</td>
</tr>
<tr>
<td>Logic trigger pattern time qualification</td>
<td>TDS 510A, 500B, 600B, &amp; 700A:</td>
</tr>
<tr>
<td>Lower limit</td>
<td>5 ns</td>
</tr>
<tr>
<td>Upper limit</td>
<td>5 ns</td>
</tr>
<tr>
<td>Logic trigger Setup/Hold times</td>
<td>TDS 500B, 600B &amp; 700A: 3 ns</td>
</tr>
<tr>
<td>Setup</td>
<td>TDS 500B, 600B &amp; 700A: 2 ns</td>
</tr>
<tr>
<td>Hold</td>
<td></td>
</tr>
<tr>
<td>Logic trigger sources and levels (Setup/Hold)</td>
<td>TDS 500B, 600B &amp; 700A: Data Source = Channel 1 = 1.4 V</td>
</tr>
<tr>
<td></td>
<td>Clock Source = Channel 2 = 1.4 V</td>
</tr>
<tr>
<td></td>
<td>(Source levels are clipped to 1.2 V at the default volts/division setting established by Factory Init)</td>
</tr>
<tr>
<td></td>
<td>Clock Edge = Rising</td>
</tr>
<tr>
<td>Logic trigger threshold (all channels) (pattern</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 1.4 V</td>
</tr>
<tr>
<td>and state)</td>
<td>(clipped to 1.2 V at the default volts/division setting when no 10X probe attached)</td>
</tr>
<tr>
<td>Logic trigger triggers when ... (pattern and</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Goes TRUE</td>
</tr>
<tr>
<td>state)</td>
<td></td>
</tr>
<tr>
<td>Main trigger mode</td>
<td>Auto</td>
</tr>
<tr>
<td>Main trigger type</td>
<td>Edge</td>
</tr>
<tr>
<td>Math1 definition</td>
<td>Ch 1 + Ch 2</td>
</tr>
<tr>
<td>Math1 extended processing</td>
<td>TDS 510A, 500B, 600B &amp; 700A: No extended processing</td>
</tr>
<tr>
<td>Math2 definition</td>
<td>Ch 1 – Ch 2 (FFT of Ch 1 for instruments with Option 2F Advanced DSP Math)</td>
</tr>
<tr>
<td>Math2 extended processing</td>
<td>TDS 510A, 500B, 600B &amp; 700A: No extended processing</td>
</tr>
<tr>
<td>Math3 definition</td>
<td>Inv of Ch 1</td>
</tr>
<tr>
<td>Math3 extended processing</td>
<td>TDS 510A, 500B, 600B &amp; 700A: No extended processing</td>
</tr>
<tr>
<td>Measure delay edges</td>
<td>Both rising and forward searching</td>
</tr>
<tr>
<td>Measure delay to</td>
<td>Channel 1 (Ch1)</td>
</tr>
</tbody>
</table>
### Table D–1: Factory Initialization Settings (Cont.)

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure gating</td>
<td>Off</td>
</tr>
<tr>
<td>Measure high ref</td>
<td>90% and 0 V (units)</td>
</tr>
<tr>
<td>Measure high-low setup</td>
<td>Histogram</td>
</tr>
<tr>
<td>Measure low ref</td>
<td>10% and 0 V (units)</td>
</tr>
<tr>
<td>Measure mid ref</td>
<td>50% and 0 V (units)</td>
</tr>
<tr>
<td>Measure mid2 ref</td>
<td>50% and 0 V (units)</td>
</tr>
<tr>
<td>Pulse glitch filter state</td>
<td>TDS 510A, 500B, 600B &amp; 700A: On (Accept glitch)</td>
</tr>
<tr>
<td>Pulse glitch trigger polarity</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Positive</td>
</tr>
<tr>
<td>Pulse glitch width</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 2.0 ns</td>
</tr>
<tr>
<td>Pulse runt high threshold</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 1.2 V</td>
</tr>
<tr>
<td>Pulse runt low threshold</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 0.8 V</td>
</tr>
<tr>
<td>Pulse runt trigger polarity</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Positive</td>
</tr>
<tr>
<td>Pulse runt triggers when ...</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Occurs</td>
</tr>
<tr>
<td>Pulse runt width</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 2.0 ns</td>
</tr>
<tr>
<td>Pulse slew rate delta time</td>
<td>2.0 ns</td>
</tr>
<tr>
<td>Pulse slew rate polarity</td>
<td>Positive</td>
</tr>
<tr>
<td>Pulse slew rate thresholds</td>
<td>Trig if faster than 1.80 V</td>
</tr>
<tr>
<td></td>
<td>800 mV</td>
</tr>
<tr>
<td>Pulse slew rate triggers when ...</td>
<td>Trig if faster than</td>
</tr>
<tr>
<td>Pulse timeout polarity</td>
<td>TDS 500B &amp; 600B: Either</td>
</tr>
<tr>
<td>Pulse timeout time</td>
<td>TDS 500B &amp; 600B: 2.0 ns</td>
</tr>
<tr>
<td>Pulse trigger class</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Glitch</td>
</tr>
<tr>
<td>Pulse trigger level</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 0.0 V</td>
</tr>
<tr>
<td>Pulse trigger source (Glitch, runt, and width, and timeout)</td>
<td>TDS 500B, 600B, &amp; 700A: Channel 1 (Ch1)</td>
</tr>
<tr>
<td>Pulse width lower limit</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 2.0 ns</td>
</tr>
<tr>
<td>Pulse width trigger polarity</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Positive</td>
</tr>
<tr>
<td>Pulse width trigger when</td>
<td>TDS 510A, 500B, 600B &amp; 700A: Within limits</td>
</tr>
<tr>
<td>Pulse width upper limit</td>
<td>TDS 510A, 500B, 600B &amp; 700A: 2.0 ns</td>
</tr>
<tr>
<td>Repetitive signal</td>
<td>TDS 500B &amp; 700A: On</td>
</tr>
<tr>
<td>RS-232 parameters</td>
<td>No change</td>
</tr>
<tr>
<td>Saved setups</td>
<td>No change</td>
</tr>
</tbody>
</table>
### Table D–1: Factory Initialization Settings (Cont.)

<table>
<thead>
<tr>
<th>Control</th>
<th>Changed by Factory Init to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saved waveforms</td>
<td>No change</td>
</tr>
<tr>
<td>Stop after</td>
<td>R/S button</td>
</tr>
<tr>
<td>Vertical bandwidth (all channels)</td>
<td>Full</td>
</tr>
<tr>
<td>Vertical coupling (all channels)</td>
<td>DC</td>
</tr>
<tr>
<td>Vertical impedance (termination) (all channels)</td>
<td>1 MΩ</td>
</tr>
<tr>
<td>Vertical offset (all channels)</td>
<td>0 V</td>
</tr>
<tr>
<td>Vertical position (all channels)</td>
<td>0 divs.</td>
</tr>
<tr>
<td>Vertical volts per division (all channels)</td>
<td>100 mV per division</td>
</tr>
<tr>
<td>Zoom dual</td>
<td>TDS 400A, 500B, 600B, &amp; 700A: Off</td>
</tr>
<tr>
<td>Zoom dual offset</td>
<td>TDS 400A, 500B, 600B, &amp; 700A: Minimum available</td>
</tr>
<tr>
<td>Zoom, dual window, selected graticule</td>
<td>TDS 400A, 500B, 600B, &amp; 700A: Upper</td>
</tr>
<tr>
<td>Zoom graticule</td>
<td>Upper</td>
</tr>
<tr>
<td>Zoom horizontal (all channels)</td>
<td>TDS 510A: 1.0X</td>
</tr>
<tr>
<td></td>
<td>TDS 400A, 500B, 600B, 700A: 2.0X</td>
</tr>
<tr>
<td>Zoom horizontal lock</td>
<td>All</td>
</tr>
<tr>
<td>Zoom horizontal position (all channels)</td>
<td>50% = 0.5 (the middle of the display)</td>
</tr>
<tr>
<td>Zoom mode</td>
<td>Off</td>
</tr>
<tr>
<td>Zoom vertical (all channels)</td>
<td>TDS 510A: 1.0X</td>
</tr>
<tr>
<td></td>
<td>TDS 400A, 500B, 600B, 700A: 2.0X</td>
</tr>
<tr>
<td>Zoom vertical position (all channels)</td>
<td>0 divisions</td>
</tr>
</tbody>
</table>
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 diagrams. The syntax diagrams in this manual use BNF notation.

Controller
A computer or other device that sends commands to and accepts responses from the digitizing 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 can be either an EOI or the ASCII code for line feed (LF).

Equivalent-time sampling (ET)
A sampling mode in which the oscilloscope acquires signals over many repetitions of the event. The TDS Family Series Digitizing Oscilloscopes use a type of equivalent time sampling called random equivalent time sampling. It utilizes an internal clock that runs asynchronously with respect to the input signal and the signal trigger. The oscilloscope takes samples continuously, independent of the trigger position, and displays them based on the time difference between the sample and the trigger. Although the samples are taken sequentially in time, they are random with respect to the trigger.

Real-time sampling
A sampling mode where the digitizing oscilloscope samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

GPIB
Acronym for General Purpose Interface Bus, the common name for the communications interface system defined in IEEE Std 488.
IEEE
Acronym for the Institute for Electrical and Electronic 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 C.

TEKSecure
A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.
Index

A

Abbreviating, command, 2–4
ACQUIRE?, 2–39
ACQUIRE:AUTOSAVE, 2–39
ACQUIRE:MODE, 2–40
ACQUIRE:NUMACQ?, 2–42
ACQUIRE:NUMAVG, 2–43
ACQUIRE:NUMENV, 2–43
ACQUIRE:REPEAT, 2–44
ACQUIRE:STATE, 2–45
ACQUIRE:STOPAFTER, 2–46
Acquisition command group, 2–11, 2–12
Acquisition commands
ACQUIRE?, 2–39
ACQUIRE:AUTOSAVE, 2–39
ACQUIRE:MODE, 2–40
ACQUIRE:NUMACQ?, 2–42
ACQUIRE:NUMAVG, 2–43
ACQUIRE:NUMENV, 2–43
ACQUIRE:REPEAT, 2–44
ACQUIRE:STATE, 2–45
ACQUIRE:STOPAFTER, 2–46
Address, Definition of, Glossary–1
ALIAS, 2–47
Alias commands
ALIAS, 2–47
ALIAS:CATALOG?, 2–48
ALIAS:DEFINE, 2–48
ALIAS:DELETE, 2–50
ALIAS:DELETE:ALL, 2–50
ALIAS:DELETE:NAME, 2–50
ALIAS:STATE, 2–51
ALIAS:CATALOG?, 2–48
ALIAS:DEFINE, 2–48
ALIAS:DELETE, 2–50
ALIAS:DELETE:ALL, 2–50
ALIAS:DELETE:NAME, 2–50
ALIAS:STATE, 2–51
CH<x>?, 2–61
CH<x>:BANDWIDTH, 2–61
CH<x>:COUPLING, 2–62
CH<x>:DESKEW, 2–63
CH<x>:IMPEDANCE, 2–64
CH<x>:OFFSET, 2–64
CH<x>:POSITION, 2–66
CH<x>:PROBE?, 2–66
CH<x>:PROBECAL?, 2–67
CH<x>:SCALE, 2–67
CH<x>:VOLTS, 2–68
Channel, command mnemonic, 2–7
CH<x>, command mnemonic, 2–7
APPMENU, 2–55
APPMENU:LABEL, 2–56
APPMENU:LABEL:BOTTOM, 2–56
APPMENU:LABEL:RIGHT, 2–57
Argument, command, 2–2
ASCII, 2–1, Glossary–1
Audio indicator, 2–59
AUTOSET, 2–59

B

Backus-Naur Form, 2–1
Definition of, Glossary–1
BELL, 2–59
Block, command argument, 2–9
BNF, Glossary–1
BNF (Backus-Naur form), 2–1
Break, 2–3
BUSY?, 2–60

C

*CAL?, 2–60
Calibration and diagnostic command group, 2–13
Calibration and diagnostic commands
*CAL?, 2–60
DIAG:RESULT:FLAG?, 2–98
DIAG:RESULT:LOG?, 2–99
DIAG:SELECT:ACQUISITION, 2–100
DIAG:SELECT:ALL, 2–100
DIAG:SELECT:CPU, 2–100
DIAG:SELECT:DISPLAY, 2–101
DIAG:SELECT:FPANEL, 2–101
DIAG:STATE, 2–102
CH<x>?, 2–61
CH<x>:BANDWIDTH, 2–61
CH<x>:COUPLING, 2–62
CH<x>:DESKEW, 2–63
CH<x>:IMPEDANCE, 2–64
CH<x>:OFFSET, 2–64
CH<x>:POSITION, 2–66
CH<x>:PROBE?, 2–66
CH<x>:PROBECAL?, 2–67
CH<x>:SCALE, 2–67
CH<x>:VOLTS, 2–68
Channel, command mnemonic, 2–7
CH<x>, command mnemonic, 2–7

TDS Family Oscilloscope Programmer Manual

Index–1
Clear Status, 2–69
CLEARMENU, 2–69
*CLS, 2–69

Command
Abbreviating, 2–4
Argument, 2–2
Block argument, 2–9
Common, 2–21, 2–23
Concatenating, 2–4
Header, 2–2
Message, 2–2
Mnemonic, 2–2
Query, 2–1
Rules for forming, 2–1
Separator, 2–2
Set, 2–1
Syntax, 2–1
BNF (Backus-Naur form), 2–1

Command argument
Numeric, 2–7
Quoted string, 2–8

Command Descriptions, 1–1

Command group
Acquisition, 2–11, 2–12
Application menu, 2–12
Calibration and diagnostic, 2–13
Cursor, 2–13
Display, 2–14
File system, 2–16
Hardcopy, 2–17
Horizontal, 2–17
Limit test, 2–19
Measurement, 2–19
Miscellaneous, 2–21
RS-232, 2–22
Save and recall, 2–22
Status and error, 2–23
Trigger, 2–24
Vertical, 2–31
Waveform, 2–32
Zoom, 2–37

Command Groups, 1–1

Command mnemonic
Channel, 2–7
CH<x>, 2–7
Cursor position, 2–6
Math waveform, 2–7
MATH<x>, 2–7
Measurement specifier, 2–6
MEAS<x>, 2–6
POSITION<x>, 2–6
Reference waveform, 2–7
REF<x>, 2–7

Waveform, 2–7
<wfm>, 2–7

Command syntax, 2–1
BNF (Backus-Naur form), 2–1

Commands, 1–1
Parts of, 1–1
Syntax diagrams, 1–1
Common command, 2–21, 2–23
Common GPIB commands
*CAL?, 2–60
*CLS, 2–69

Concatenating, command, 2–4
Configuration, Command query, 2–158
Controller, Definition of, Glossary–1
Cursor command group, 2–13

Cursor commands
CURSOR?, 2–70
CURSOR:FUNCTION, 2–70
CURSOR:HBARS?, 2–71
CURSOR:HBARS:DELTA?, 2–71
CURSOR:HBARS:POSITION<x>, 2–72
CURSOR:HBARS:POSITION<x>PCNT, 2–72
CURSOR:HBARS:SELECT, 2–73
CURSOR:HBARS:UNITS, 2–74
CURSOR:MODE, 2–74
CURSOR:PAIRED, 2–75
CURSOR:PAIRED:DELTA, 2–76
CURSOR:PAIRED:HPOS1, 2–76
CURSOR:PAIRED:HPOS2, 2–77
CURSOR:PAIRED:POSITION<x>, 2–77
CURSOR:PAIRED:POSITION<x>PCNT, 2–78
CURSOR:PAIRED:SELECT, 2–78
CURSOR:PAIRED:UNITS, 2–79
CURSOR:PAIRED:VDELTA, 2–80
CURSOR:VBARS, 2–80
CURSOR:VBARS:DELTA?, 2–81
CURSOR:VBARS:POSITION<x>, 2–82
CURSOR:VBARS:POSITION<x>PCNT, 2–82
CURSOR:VBARS:SELECT, 2–83
CURSOR:VBARS:UNITS, 2–84
CURSOR:VBARS:UNITSTRING, 2–86

Cursor position, command mnemonic, 2–6
CURSOR?, 2–70
CURSOR:FUNCTION, 2–70
CURSOR:HBARS?, 2–71
CURSOR:HBARS:DELTA?, 2–71
CURSOR:HBARS:POSITION<x>, 2–72
CURSOR:HBARS:POSITION<x>PCNT, 2–72
CURSOR:HBARS:SELECT, 2–73
CURSOR:HBARS:UNITS, 2–74
CURSOR:MODE, 2–74
CURSOR:PAIRED, 2–75
CURSOR:PAIRED:HDELTA, 2–76
CURSOR: PAIRED: HPOS1, 2–76
CURSOR: PAIRED: HPOS2, 2–77
CURSOR: PAIRED: POSITION << PCNT, 2–78
CURSOR: PAIRED: SELECT, 2–78
CURSOR: PAIRED: UNITS, 2–79
CURSOR: V BARS, 2–80
CURSOR: V BARS: DELTA?, 2–81
CURSOR: V BARS: POSITION << PCNT, 2–82
CURSOR: V BARS: SELECT, 2–83
CURSOR: V BARS: UNITS, 2–84
CURSOR: V BARS: UNITSTRING, 2–86
CURVE, 2–86

D
DATA, 2–88
DATA: DESTINATION, 2–88
DATA: ENCDG, 2–89
DATA: SOURCE, 2–91
DATA: START, 2–92
DATA: STOP, 2–93
DATA: TARGET, 2–93
DATA: WIDTH, 2–94
DATE, 2–95
DCL, C–2
*DDT, 2–95
DELETE: SETUP, 2–96
DELETE: WAVEFORM, 2–97
DESE command, 2–97, 3–3
DESER register, 2–97, 2–203, 3–3
Device Clear, 2–3, C–2
DIAG: RESULT: FLAG?, 2–98
DIAG: RESULT: LOG?, 2–99
DIAG: SELECT: ACQUISITION, 2–100
DIAG: SELECT: ALL, 2–100
DIAG: SELECT: CPU, 2–100
DIAG: SELECT: DISPLAY, 2–101
DIAG: SELECT: FPANEL, 2–101
DIAG: STATE, 2–102
Diagram, syntax, 2–10
Disks included with this manual, 1–2
Display command group, 2–14
Display commands
  CLEARMENU, 2–69
  DISPLAY?, 2–103
  DISPLAY: CLOCK, 2–103
  DISPLAY: COLOR: CONTRAST, 2–104
  DISPLAY: COLOR: MAP: item name: TO, 2–106
  DISPLAY: COLOR: MAP: item: BYCONTENTS, 2–105
  DISPLAY: COLOR: PALETTE: palette name: item name, 2–109
  DISPLAY: COLOR: PALETTE: palette name: RESET, 2–108
  DISPLAY: COLOR: PALETTE: PERSISTENCE, 2–107
  DISPLAY: COLOR: PALETTE: REGULAR, 2–107
  DISPLAY: COLOR: PALETTE: RESETALL, 2–108
  DISPLAY: DATE/TIME. See DISPLAY: CLOCK
  DISPLAY: FILTER, 2–111
  DISPLAY: FORMAT, 2–112
  DISPLAY: GRATICULE, 2–113
  DISPLAY: INSTAVU: PERSISTENCE, 2–114
  DISPLAY: INSTAVU: STYLE, 2–115
  DISPLAY: INSTAVU: VARPERSIST, 2–115
  DISPLAY: INTENSITY?, 2–116
  DISPLAY: INTENSITY: CONTRAST, 2–116
  DISPLAY: INTENSITY: OVERALL, 2–117
  DISPLAY: INTENSITY: TEXT, 2–118
  DISPLAY: INTENSITY: WAVEFORM, 2–118
  DISPLAY: MODE, 2–119
  DISPLAY: PERSISTENCE, 2–119
  DISPLAY: STYLE, 2–120
  DISPLAY: TRIGBAR, 2–121
  DISPLAY: TRIGT, 2–122
  MESSAGE, 2–197
  MESSAGE: BOX, 2–197
  MESSAGE: SHOW, 2–199
  MESSAGE: STATE, 2–200
  DISPLAY?, 2–103
  DISPLAY: CLOCK, 2–103
  DISPLAY: COLOR: CONTRAST, 2–104
  DISPLAY: COLOR: MAP: item name: TO, 2–106
  DISPLAY: COLOR: MAP: item: BYCONTENTS, 2–105
  DISPLAY: COLOR: PALETTE: palette name: item name, 2–109
  DISPLAY: COLOR: PALETTE: palette name: RESET, 2–108
  DISPLAY: COLOR: PALETTE: PERSISTENCE, 2–107
  DISPLAY: COLOR: PALETTE: REGULAR, 2–107
  DISPLAY: COLOR: PALETTE: RESETALL, 2–108
  DISPLAY: DATE/TIME. See DISPLAY: CLOCK
  DISPLAY: FILTER, 2–111
  DISPLAY: FORMAT, 2–112
  DISPLAY: GRATICULE, 2–113
  DISPLAY: INSTAVU: PERSISTENCE, 2–114
  DISPLAY: INSTAVU: STYLE, 2–115
  DISPLAY: INSTAVU: VARPERSIST, 2–115
  DISPLAY: INTENSITY?, 2–116
DISPLAY:INTENSITY:CONTRAST, 2–116
DISPLAY:INTENSITY:OVERALL, 2–117
DISPLAY:INTENSITY:WAVEFORM, 2–118
DISPLAY:MODE, 2–119
DISPLAY:PERSISTENCE, 2–119
DISPLAY:STYLE, 2–120
DISPLAY:TRIGBAR, 2–121
DISPLAY:TRIGT, 2–122

FILESYSTEM:COPY, 2–127
FILESYSTEM:CWD, 2–128
FILESYSTEM:DELETE, 2–128
FILESYSTEM:DELWARN, 2–129
FILESYSTEM:DIR, 2–130
FILESYSTEM:FORMAT, 2–130
FILESYSTEM:FREESPACE, 2–130
FILESYSTEM:MKDIR, 2–131
FILESYSTEM:OVERWRITE, 2–131
FILESYSTEM:PRINT, 2–132
FILESYSTEM:RENAMEN, 2–133
FILESYSTEM:RMDIR, 2–133
FITTOSCREEN, 2–150

GET, C–2
Glitch trigger, 2–250, 2–251, 2–252, 2–253
Go to local, C–2
GPIB, Glossary–1
Configurations, 1–4
Connection rules, 1–4
EOM (end of message), 2–5
Function subsets, C–1
Group execute trigger, C–2
GTL, C–2

HARDCOPY, 2–134
Hardcopy command group, 2–17
Hardcopy commands
HARDCOPY, 2–134
HARDCOPY:FILENAME, 2–135
HARDCOPY:FORMAT, 2–136
HARDCOPY:LAYOUT, 2–138
HARDCOPY:PALETTE, 2–138
HARDCOPY:PORT, 2–139
HARDCOPY:FILENAME, 2–135
HARDCOPY:FORMAT, 2–136
HARDCOPY:LAYOUT, 2–138
HARDCOPY:PALETTE, 2–138
HARDCOPY:PORT, 2–139
HDR, 2–140
HEADER, 2–140
Header
Command, 2–2, 2–140
Included in query response, 2–140, 2–290
Horizontal command group, 2–17
Horizontal commands
HORIZONTAL?, 2–141
HORIZONTAL:CLOCK, 2–142
HORIZONTAL:CLOCK:MAXRATE, 2–142
HORIZONTAL:DELAY?, 2–143
HORIZONTAL:DELAY:MODE, 2–143
HORIZONTAL:DELAY:SCALE, 2–144
HORIZONTAL:SECDIV, 2–145
HORIZONTAL:DELAY:TIME, 2–145
HORIZONTAL:DELAY:TIME?, 2–146
HORIZONTAL:DELAY:TIME:RUNSAFTER, 2–146
HORIZONTAL:DELAY:TIME:TRIGAFTER, 2–147
HORIZONTAL:FASTFRAME:COUNT, 2–148
HORIZONTAL:FASTFRAME:LENGTH, 2–148
HORIZONTAL:FASTFRAME:POSITION, 2–149
HORIZONTAL:FITTOSCREEN, 2–150
HORIZONTAL:MAIN?, 2–151
HORIZONTAL:MAIN:SCALE, 2–151
HORIZONTAL:MAIN:SECDIV, 2–152
HORIZONTAL:MODE, 2–153
HORIZONTAL:POSITION, 2–154
HORIZONTAL:RECORDLENGTH, 2–154
HORIZONTAL:ROLL, 2–155
HORIZONTAL:SCALE, 2–156
HORIZONTAL:TRIGGER?, 2–156
HORIZONTAL:TRIGGER:POSITION, 2–157
HORIZONTAL?, 2–141
HORIZONTAL:CLOCK, 2–142
HORIZONTAL:CLOCK:MAXRATE, 2–142
HORIZONTAL:DELAY?, 2–143
HORIZONTAL:DELAY:MODE, 2–143
HORIZONTAL:DELAY:SCALE, 2–144
HORIZONTAL:SECDIV, 2–145
HORIZONTAL:DELAY:TIME, 2–145
HORIZONTAL:DELAY:TIME?, 2–146
HORIZONTAL:DELAY:TIME:RUNSAFTER, 2–146
HORIZONTAL:DELAY:TIME:TRIGAFTER, 2–147
HORIZONTAL:FASTFRAME:COUNT, 2–148
HORIZONTAL:FASTFRAME:LENGTH, 2–148
HORIZONTAL:FASTFRAME:POSITION, 2–149
HORIZONTAL:FASTFRAME:STATE, 2–149
HORIZONTAL:FITTOSCREEN, 2–150
HORIZONTAL:MODE, 2–153
HORIZONTAL:POSITION, 2–154
HORIZONTAL:RECORDLENGTH, 2–154
HORIZONTAL:ROLL, 2–155
HORIZONTAL:SCALE, 2–156
HORIZONTAL:SECDIV, 2–156
HORIZONTAL:TRIGGER?, 2–156
HORIZONTAL:TRIGGER:POSITION, 2–157
HORIZONTAL?, 2–141
HORIZONTAL:CLOCK, 2–142
HORIZONTAL:CLOCK:MAXRATE, 2–142
HORIZONTAL:DELAY?, 2–143
HORIZONTAL:DELAY:MODE, 2–143
HORIZONTAL:DELAY:SCALE, 2–144
HORIZONTAL:SECDIV, 2–145
HORIZONTAL:DELAY:TIME, 2–145
HORIZONTAL:DELAY:TIME?, 2–146
HORIZONTAL:DELAY:TIME:RUNSAFTER, 2–146
HORIZONTAL:DELAY:TIME:TRIGAFTER, 2–147
HORIZONTAL:FASTFRAME:COUNT, 2–148
HORIZONTAL:FASTFRAME:LENGTH, 2–148
HORIZONTAL:FASTFRAME:POSITION, 2–149
HORIZONTAL:FASTFRAME:STATE, 2–149
HORIZONTAL:FITTOSCREEN, 2–150
HORIZONTAL:MAIN?, 2–151
HORIZONTAL:MAIN:SCALE, 2–151
HORIZONTAL:MAIN:SECDIV, 2–152
HORIZONTAL:MODE, 2–153
HORIZONTAL:POSITION, 2–154
HORIZONTAL:RECORDLENGTH, 2–154
HORIZONTAL:ROLL, 2–155
HORIZONTAL:SCALE, 2–156
HORIZONTAL:SECDIV, 2–156
HORIZONTAL:TRIGGER?, 2–156
HORIZONTAL:TRIGGER:POSITION, 2–157
HORIZONTAL?, 2–141
HORIZONTAL:CLOCK, 2–142
HORIZONTAL:CLOCK:MAXRATE, 2–142
HORIZONTAL:DELAY?, 2–143
HORIZONTAL:DELAY:MODE, 2–143
HORIZONTAL:DELAY:SCALE, 2–144
HORIZONTAL:SECDIV, 2–145
HORIZONTAL:DELAY:TIME, 2–145
HORIZONTAL:DELAY:TIME?, 2–146
HORIZONTAL:DELAY:TIME:RUNSAFTER, 2–146
HORIZONTAL:DELAY:TIME:TRIGAFTER, 2–147
HORIZONTAL:FASTFRAME:COUNT, 2–148
HORIZONTAL:FASTFRAME:LENGTH, 2–148
HORIZONTAL:FASTFRAME:POSITION, 2–149
HORIZONTAL:FASTFRAME:STATE, 2–149
HORIZONTAL:FITTOSCREEN, 2–150
HORIZONTAL:MODE, 2–153
HORIZONTAL:POSITION, 2–154
HORIZONTAL:RECORDLENGTH, 2–154
HORIZONTAL:ROLL, 2–155
HORIZONTAL:SCALE, 2–156
HORIZONTAL:SECDIV, 2–156
HORIZONTAL:TRIGGER?, 2–156
HORIZONTAL:TRIGGER:POSITION, 2–157

I
ID? , 2–157
*IDN?, 2–158
IEEE, Glossary–2
IEEE Std 488.2-1987, 1–3, 2–1, 2–21, 2–23
Instrument setup, 1–3
Interface message, C–2

L
Limit test command group, 2–19
Limit Test commands
LIMIT:BELL, 2–158
LIMIT:COMPARE:CH<x>, 2–159
LIMIT:COMPARE:MAThx , 2–160
LIMIT:HARDCOPY, 2–161
LIMIT:STATE, 2–162
LIMIT:TEMPLATE, 2–162
LIMIT:TEMPLATE:DESTINATION, 2–163
LIMIT:TEMPLATE:SOURCE, 2–164
LIMIT:TEMPLATE:TOLERANCE: VERTICAL, 2–165
LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL, 2–164
LIMIT:BELL, 2–158
LIMIT:COMPARE:CH<X>, 2–159
LIMIT:COMPARE:MAT<XX>, 2–160
LIMIT:HARDCOPY, 2–161
LIMIT:STATE, 2–162
LIMIT:TEMPLATE, 2–162
LIMIT:TEMPLATE:DESTINATION, 2–163
LIMIT:TEMPLATE:SOURCE, 2–164
LIMIT:TEMPLATE:TOLERANCE: VERTICAL, 2–165
LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL, 2–164
LLO, C–2
Local lock out, C–2
LOCK, 2–166
*LRN?, 2–167

M
Manual trigger, Simulation with command, 2–288
Math waveform, command mnemonic, 2–7
MATH<x?>, 2–168
MATH<xx>:DEFINE, 2–168

TDS Family Oscilloscope Programmer Manual
Index

MA T H<x>:NUMAVg, 2–170
MA T H<x>:PROCe ssing, 2–170
MA T H<x>, command mnemonic, 2–7
Measurement command group, 2–19
Measurement commands
  MEASUREMENT?, 2–171
  MEASUREMENT:CLEARSNAPSHOT, 2–172
  MEASUREMENT:GATING, 2–172
  MEASUREMENT:IMMED?, 2–173
  MEASUREMENT:IMMED:DELAY?, 2–173
  MEASUREMENT:IMMED:DELAY:DIRECTION, 2–174
  MEASUREMENT:IMMED:DELAY:EDGE1, 2–174
  MEASUREMENT:IMMED:DELAY:EDGE2, 2–175
  MEASUREMENT:IMMED:SOURCE1, 2–176
  MEASUREMENT:IMMED:SOURCE2, 2–177
  MEASUREMENT:IMMED:TYPE, 2–177
  MEASUREMENT:IMMED:UNITS?, 2–180
  MEASUREMENT:IMMED:VALUE?, 2–181
  MEASUREMENT:MEAS<x>?, 2–181
  MEASUREMENT:MEAS<x>: DELAY?, 2–181
  MEASUREMENT:MEAS<x>: DELAY:DIRECTION?, 2–182
  MEASUREMENT:MEAS<x>: DELAY:EDGE1, 2–182
  MEASUREMENT:MEAS<x>: DELAY:EDGE2, 2–183
  MEASUREMENT:MEAS<x>: DELAY:SOURCE1, 2–184
  MEASUREMENT:MEAS<x>: DELAY:SOURCE2, 2–185
  MEASUREMENT:MEAS<x>: STATE, 2–186
  MEASUREMENT:MEAS<x>: TYPE, 2–186
  MEASUREMENT:MEAS<x>: UNITS?, 2–189
  MEASUREMENT:MEAS<x>: VALUES?, 2–189
  MEASUREMENT:METHOD, 2–190
  MEASUREMENT:REFLEVEL?, 2–190
  MEASUREMENT:REFLEVEL:ABSOLUTE:HIG H, 2–191
  MEASUREMENT:REFLEVEL:ABSOLUTE:MID, 2–192
  MEASUREMENT:REFLEVEL:ABSOLUTE:MID2, 2–193
  MEASUREMENT:REFLEVEL:METHOD, 2–193
  MEASUREMENT:REFLEVEL:PERCENT:HIG H, 2–194
  MEASUREMENT:REFLEVEL:PERCENT:LOW, 2–195
  MEASUREMENT:REFLEVEL:PERCENT:MID, 2–195
  MEASUREMENT:REFLEVEL:PERCENT:MID2, 2–196
  MEASUREMENT:SNAPSHOT, 2–197

Measurement specifier, command mnemonic, 2–6
MEASUREMENT?, 2–171
MEASUREMENT:CLEARSNAPSHOT, 2–172
MEASUREMENT:GATING, 2–172
MEASUREMENT:IMMED?, 2–173
MEASUREMENT:IMMED:DELAY?, 2–173
MEASUREMENT:IMMED:DELAY:DIRECTION, 2–174
MEASUREMENT:IMMED:DELAY:EDGE1, 2–174
MEASUREMENT:IMMED:DELAY:EDGE2, 2–175
MEASUREMENT:IMMED:SOURCE1, 2–176
MEASUREMENT:IMMED:SOURCE2, 2–177
MEASUREMENT:IMMED:TYPE, 2–177
MEASUREMENT:IMMED:UNITS?, 2–180
MEASUREMENT:IMMED:VALUE?, 2–181
MEASUREMENT:MEAS<x>?, 2–181
MEASUREMENT:MEAS<x>: DELAY?, 2–181
MEASUREMENT:MEAS<x>: DELAY:DIRECTION?, 2–182
MEASUREMENT:MEAS<x>: DELAY:EDGE1, 2–182
MEASUREMENT:MEAS<x>: DELAY:EDGE2, 2–183
MEASUREMENT:MEAS<x>: DELAY:SOURCE1, 2–184
MEASUREMENT:MEAS<x>: DELAY:SOURCE2, 2–185
MEASUREMENT:MEAS<x>: STATE, 2–186
MEASUREMENT:MEAS<x>: TYPE, 2–186
MEASUREMENT:MEAS<x>: UNITS?, 2–189
MEASUREMENT:MEAS<x>: VALUES?, 2–189
MEASUREMENT:METHOD, 2–190
MEASUREMENT:REFLEVEL?, 2–190
MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH, 2–191
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW, 2–191
MEASUREMENT:REFLEVEL:ABSOLUTE:MID, 2–192
MEASUREMENT:REFLEVEL:ABSOLUTE:MID2, 2–193
MEASUREMENT:REFLEVEL:METHOD, 2–193
MEASUREMENT:REFLEVEL:PERCENT:HIGH, 2–194
MEASUREMENT:REFLEVEL:PERCENT:LOW, 2–195
MEASUREMENT:REFLEVEL:PERCENT:MID, 2–195
MEASUREMENT:REFLEVEL:PERCENT:MID2, 2–196
MEASUREMENT:SNAPSHOT, 2–197
MEAS<x>, command mnemonic, 2–6
MESSAGE, 2–197
Message
  Command, 2–2
  Command terminator, 2–5
  Handling, 3–1
  Table of program messages, 3–12
MESSAGE:BOX, 2–197
MESSAGE:SHOW, 2–199
MESSAGE:STATE, 2–200
Messages, Status and error, 1–2
Miscellaneous, LOCK, 2–166
Miscellaneous command group, 2–21
Miscellaneous commands
  AUTOSET, 2–59
  BELL, 2–59
  DATE, 2–95
  *DDT, 2–95
  FACTORY, 2–126
  HDR, 2–140
  HEADER, 2–140
  *IDN?, 2–158
  *LRN?, 2–167
  NEWPASS, 2–200
  PASSWORD, 2–202
  *PUD, 2–204
  REM, 2–207
  SET, 2–217
  TEKSECURE, 2–219
  TIME, 2–220
  *TRG, 2–288
  UNLOCK, 2–289
  VERBOSE, 2–290
Mnemonic, command, 2–2

N
NEWPASS, 2–200
Numeric, command argument, 2–7

O
*OPC, 2–201
  Operation complete command, 2–201
  Operation complete wait, 2–290
*OPT, 2–202
  Option Identification Query, 2–202
  Output queue, 3–5

P
Parallel poll, C–2
Parts of commands, 1–1
PASSWORD, 2–202
Pattern trigger, 2–238, 2–239, 2–240, 2–241
POSITION<x>, command mnemonic, 2–6
Power-on status clear command, 2–203
PPC, C–2
PPD, C–2
PPE, C–2
PPU, C–2
Programming Examples, 1–2, 4–1
*PSC, 2–203
*PSC command, 3–4
*PUD, 2–204

Q
Query, Header in query response, 2–140, 2–290
Query command, 2–1
Queue
  Event, 3–5
  Output, 3–5
QuickBASIC, 4–1, Glossary–2
QuickC, 4–1, Glossary–2
Quoted string, command argument, 2–8

R
*RCL, 2–205
  Real-time sampling, Glossary–1
Recall setting command, 2–205
RECALL:SETUP, 2–205
RECALL:WAVEFORM, 2–206
Reference waveform, command mnemonic, 2–7
REF<x>, command mnemonic, 2–7
Register
  DESER, 2–97, 2–203, 3–3
  ESER, 2–122, 2–203, 3–3
  SBR, 2–219, 3–2
  SESR, 2–69, 2–123, 2–201, 3–1
  SREER, 2–203, 2–218, 3–4
  REM, 2–207
Reset
   Command, 2–207
   Factory, 2–126
RS-232 command group, 2–22
RS-232 commands
   RS232?, 2–211
   RS232:BAUD, 2–208
   RS232:HARDFLAGGING, 2–209
   RS232:PARITY, 2–209
   RS232:SOFTWARELING, 2–210
   RS232:STOPBITS, 2–211
RS232?, 2–211
RS232: BAUD, 2–208
RS232: HARDFLAGGING, 2–209
RS232: PARITY, 2–209
RS232: SOFTWARELING, 2–210
RS232: STOPBITS, 2–211
*RST, 2–207
Rules, command forming, 2–1

S
*SAY, 2–212
Save and recall command group, 2–22
Save and recall commands
   ALLOCATE?, 2–52
   ALLOCATE:WAVEFORM?, 2–53
   ALLOCATE:WAVEFORM:FREE?, 2–53
   ALLOCATE:WAVEFORM:REF<x>, 2–54
   DELETE:SETUP, 2–96
   DELETE:WAVEFORM, 2–97
   *RCL, 2–205
   RECALL:SETUP, 2–205
   RECALL:WAVEFORM, 2–206
   *SAV, 2–212
   SAVE:SETUP, 2–212
   SAVE:WAVEFORM, 2–213
   SAVE:WAVEFORM:FILEFORMT, 2–214
Save setting command, 2–212
SAVE:SETUP, 2–212
SAVE:WAVEFORM, 2–213
SAVE:WAVEFORM:FILEFORMT, 2–214
SBR register, 2–219, 3–2
SDC, C–2
SELECT?, 2–215
SELECT:<wfim>, 2–216
SELECT:CONTROL?, 2–217
Selected device clear, C–2
Self test, 2–289
Separator, command, 2–2

Serial poll, 3–2
   Disable, C–2
   Enable, C–2
Service request enable command, 2–218
Service request enable register, 2–218
Service Requests, 1–2
SESF register, 2–69, 2–123, 2–201, 3–1
Set command, 2–1
SET?, 2–217
Sethold trigger, 2–241, 2–242, 2–243, 2–244, 2–245
Setting
   Command query, 2–167
   Query, 2–167
   Recall command, 2–205
   Save command, 2–212
Setup, Instrument preparation, 1–3
SPD, C–2
SPE, C–2
*SRE command, 2–218, 3–4
SRER register, 2–203, 2–218, 3–4
SRQ, 1–2
State trigger, 2–246, 2–247
Status, 3–1
Status and error command group, 2–23
Status and error commands
   *OPT, 2–202
   ALLEV?, 2–51
   BUSY?, 2–60
   *CLS, 2–69
   DESE, 2–97, 3–3
   *ESE, 2–122, 3–3
   *ESR?, 2–123, 3–1
   EVENT?, 2–124
   EVMSG?, 2–124
   EVQTY?, 2–125
   ID?, 2–157
   *OPC, 2–201
   *PSC, 2–203, 3–4
   *RST, 2–207
   *SRE, 2–218, 3–4
   *STB?, 2–219, 3–2
   *TST?, 2–289
   *WAI, 2–290
Status and Events, 1–2
*STB?, 2–219
*STB? query, 3–2
Syntax
   BNF (Backus-Naur form), 2–1
   Command, 2–1
   Diagram, 2–10

Index
Syntax and Commands, 1–1
Syntax diagrams, 1–1

T
Table, programming message, 3–12
TCT, C–2
Tek Standard Codes and Formats 1989, 2–23
TEKSECURE, 2–219
TEKSecure, Glossary–2
Terminator, command message, 2–5
TIME, 2–220
Time base, Manual trigger simulation, 2–288
Timeout trigger, 2–263, 2–264
*TRG, 2–288
TRIGGER, 2–220
Trigger command group, 2–24
Trigger commands
  TRIGGER, 2–220
  TRIGGER:DELAY, 2–221
  TRIGGER:DELAY:BY, 2–222
  TRIGGER:DELAY:EDGE?, 2–223
  TRIGGER:DELAY:EDGE:COUPLING, 2–223
  TRIGGER:DELAY:EDGE:SLOPE, 2–224
  TRIGGER:DELAY:EDGE:SOURCE, 2–225
  TRIGGER:DELAY:EVENTS?, 2–226
  TRIGGER:DELAY:EVENTS:COUNT, 2–226
  TRIGGER:DELAY:LEVEL, 2–227
  TRIGGER:DELAY:TIME, 2–227
  TRIGGER:DELAY:TYPE, 2–228
  TRIGGER:MAIN, 2–229
  TRIGGER:MAIN:EDGE?, 2–229
  TRIGGER:MAIN:EDGE:COUPLING, 2–229
  TRIGGER:MAIN:EDGE:SLOPE, 2–230
  TRIGGER:MAIN:EDGE:SOURCE, 2–231
  TRIGGER:MAIN:HOLDOFF?, 2–232
  TRIGGER:MAIN:HOLDOFF:ACTUAL?, 2–232
  TRIGGER:MAIN:HOLDOFF:BY, 2–233
  TRIGGER:MAIN:HOLDOFF:TIME, 2–233
  TRIGGER:MAIN:HOLDOFF:VALUE, 2–234
  TRIGGER:MAIN:LEVEL, 2–234
  TRIGGER:MAIN:LOGIC?, 2–235
  TRIGGER:MAIN:LOGIC:CLASS, 2–236
  TRIGGER:MAIN:LOGIC:CLOCK:LEVEL, 2–242
  TRIGGER:MAIN:LOGIC:DATA:LEVEL, 2–243
  TRIGGER:MAIN:LOGIC:FUNCTION, 2–236
  TRIGGER:MAIN:LOGIC:INPUT?, 2–237
  TRIGGER:MAIN:LOGIC:INPUT:CH<x>?, 2–238
  TRIGGER:MAIN:LOGIC:PATTERN: WHEN, 2–239
TRIGGER:MAIN:LOGIC:SETTHOLD:DATA:SOURCE, 2–244
TRIGGER:MAIN:LOGIC:SETTHOLD:SETTIME, 2–245
TRIGGER:MAIN:LOGIC:THRESHOLD?, 2–247
TRIGGER:MAIN:LOGIC:THRESHOLD:CH<x>, 2–248
TRIGGER:MAIN:LOGIC:WHEN, 2–248
TRIGGER:MAIN:MODE, 2–249
TRIGGER:MAIN:PULSE?, 2–249
TRIGGER:MAIN:PULSE:CLASS, 2–250
TRIGGER:MAIN:PULSE:GLITCH?, 2–251
TRIGGER:MAIN:PULSE:GLITCH:FILTER, 2–251
TRIGGER:MAIN:PULSE:GLITCH:POLARITY, 2–252
TRIGGER:MAIN:PULSE:GLITCH:WIDTH, 2–253
TRIGGER:MAIN:PULSE:RUNT?, 2–253
TRIGGER:MAIN:PULSE:RUNT:POLARITY, 2–254
TRIGGER:MAIN:PULSE:RUNT:WHEN, 2–257
TRIGGER:MAIN:PULSE:RUNT:WIDTH, 2–258
TRIGGER:MAIN:PULSE:SLEWRATE:DELTA-TIME, 2–258
TRIGGER:MAIN:PULSE:SLEWRATE:POLARITY, 2–259
TRIGGER:MAIN:PULSE:SLEWRATE:WHEN, 2–262
TRIGGER:MAIN:PULSE:SOURCE, 2–263
TRIGGER:MAIN:PULSE:TIMEOUT:POLARITY, 2–263
TRIGGER:MAIN:PULSE:WIDTH?, 2–265
TRIGGER:MAIN:PULSE:WIDTH:HIGHLIMIT, 2–265
TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT, 2–266
TRIGGER:MAIN:PULSE:WIDTH:POLARITY, 2–266
TRIGGER:MAIN:PULSE:WIDTH:WHEN, 2–267
TRIGGER:MAIN:TYPE, 2–267
TRIGGER:MAIN:VIDEO?, 2–268
TRIGGER:MAIN:VIDEO:BY, 2–269
TRIGGER:MAIN:VIDEO:FIELD, 2–269, 2–271
TRIGGER:MAIN:VIDEO:FIELDTYPE, 2–271
TRIGGER:MAIN:VIDEO:FLEXFORMAT?, 2–272
TRIGGER:MAIN:VIDEO:FLEXFORMAT:FIELD, 2–273
TRIGGER:MAIN:VIDEO:FLEXFORMAT:FRAME, 2–272
TRIGGER:MAIN:VIDEO:FLEXFORMAT:LINES, 2–274
TRIGGER:MAIN:VIDEO:FLEXFORMAT:NEGSYNCWIDTH, 2–275
TRIGGER:MAIN:VIDEO:FLEXFORMAT:VSTARTTIME, 2–275
TRIGGER:MAIN:VIDEO:FLEXFORMAT:VSTOPTIME, 2–276
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STARTTIME, 2–277
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STOPTIME, 2–277
TRIGGER:MAIN:VIDEO:HDTV, 2–278
TRIGGER:MAIN:VIDEO:HOLDOFF?, 2–279
TRIGGER:MAIN:VIDEO:HOLDOFF:VALUE, 2–279, 2–280
TRIGGER:MAIN:VIDEO:LINE, 2–280
TRIGGER:MAIN:VIDEO:LINES, 2–281
TRIGGER:MAIN:VIDEO:NTSC, 2–281
TRIGGER:MAIN:VIDEO:PAL, 2–282
TRIGGER:MAIN:VIDEO:SCAN, 2–283
TRIGGER:MAIN:VIDEO:SCANPERIOD, 2–283
TRIGGER:MAIN:VIDEO:SOURCE, 2–284
TRIGGER:MAIN:VIDEO:STANDARD, 2–284
TRIGGER:MAIN:VIDEO:SYNC, 2–285
TRIGGER:MAIN:VIDEO:SYSTEM, 2–286
TRIGGER:MAIN:VIDEO:TIME, 2–287
TRIGGER:STATE?, 2–287
TRIGGER:DELAY, 2–221
TRIGGER:DELAY:BY, 2–222
TRIGGER:DELAY:EDGE?, 2–223
TRIGGER:DELAY:EDGE:COUPLING, 2–223
TRIGGER:DELAY:EDGE:SLOPE, 2–223
TRIGGER:DELAY:EDGE:SOURCE, 2–225
TRIGGER:DELAY:EVENTS?, 2–226
TRIGGER:DELAY:EVENTS:COUNT, 2–226
TRIGGER:DELAY:LEVEL, 2–227
TRIGGER:DELAY:TIME, 2–227
TRIGGER:DELAY:TYPE, 2–228
TRIGGER:MAIN, 2–229
TRIGGER:MAIN:EDGE?, 2–229
TRIGGER:MAIN:EDGE:COUPLING, 2–229
TRIGGER:MAIN:EDGE:SLOPE, 2–230
TRIGGER:MAIN:EDGE:SOURCE, 2–231
TRIGGER:MAIN:HOLDOFF?, 2–232
TRIGGER:MAIN:HOLDOFF:ACTUAL?, 2–232
TRIGGER:MAIN:HOLDOFF:BY, 2–233
TRIGGER:MAIN:HOLDOFF:TIME, 2–233
TRIGGER:MAIN:HOLDOFF:VALUE, 2–234
TRIGGER:MAIN:LEVEL, 2–234
TRIGGER:MAIN:LOGIC?, 2–235
TRIGGER:MAIN:LOGIC:CLASS, 2–236
TRIGGER:MAIN:LOGIC:FUNCTION, 2–236
TRIGGER:MAIN:LOGIC:INPUT?, 2–237
TRIGGER:MAIN:LOGIC:INPUT:CH<x>?, 2–238
TRIGGER:MAIN:LOGIC:PATTERN:WHEN, 2–239
TRIGGER:MAIN:LOGIC:SETTHOLD:CLOCK:LEVEL, 2–242
TRIGGER:MAIN:LOGIC:SETTHOLD:DATA:SOURCE, 2–244
TRIGGER:MAIN:LOGIC:SETTHOLD:HOLDTIME, 2–245
TRIGGER:MAIN:LOGIC:SETTHOLD:SETTIME, 2–245
TRIGGER:MAIN:LOGIC:THRESHOLD?, 2–247
TRIGGER:MAIN:LOGIC:THRESHOLD<x>, 2–248
TRIGGER:MAIN:LOGIC:WHEN, 2–248
TRIGGER:MAIN:MODE, 2–249
TRIGGER:MAIN:PULSE?, 2–249
TRIGGER:MAIN:PULSE:CLASS, 2–250
TRIGGER:MAIN:PULSE:GLITCH?, 2–251
TRIGGER:MAIN:PULSE:GLITCH:FILTER, 2–251
TRIGGER:MAIN:PULSE:GLITCH:POLARITY, 2–252
TRIGGER:MAIN:PULSE:MODE, 2–253
TRIGGER:MAIN:PULSE:RUNT?, 2–253
TRIGGER:MAIN:PULSE:RUNT:POLARITY, 2–254
TRIGGER:MAIN:PULSE:RUNT:WHEN, 2–257
TRIGGER:MAIN:PULSE:WIDTH?, 2–258
TRIGGER:MAIN:PULSE:SLEWRATE:DELTA TIME, 2–258
TRIGGER:MAIN:PULSE:SLEWRATE:POLARITY, 2–259
TRIGGER:MAIN:PULSE:SOURCE, 2–261
TRIGGER:MAIN:PULSE:TIMEOUT:POLARITY, 2–263
TRIGGER:MAIN:PULSE:WIDTH?, 2–265
TRIGGER:MAIN:PULSE:WIDTH:HIGHLIMIT, 2–266
TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT, 2–266
TRIGGER:MAIN:PULSE:WIDTH:POLARITY, 2–266
TRIGGER:MAIN:PULSE:WHEN, 2–267
TRIGGER:MAIN:TYPE, 2–267
TRIGGER:MAIN:VIDEO?, 2–268
TRIGGER:MAIN:VIDEO:BY, 2–269
TRIGGER:MAIN:VIDEO:FIELD, 2–269, 2–271
TRIGGER:MAIN:VIDEO:FIELDTYPE, 2–271
TRIGGER:MAIN:VIDEO:FLEXFORMAT?, 2–272
TRIGGER:MAIN:VIDEO:FLEXFORMAT:FIELD, 2–273
TRIGGER:MAIN:VIDEO:FLEXFORMAT:FRAMERATE, 2–272
TRIGGER:MAIN:VIDEO:FLEXFORMAT:LINES, 2–274
TRIGGER:MAIN:VIDEO:FLEXFORMAT:NEGSYNCWIDTH, 2–275
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STARTTIME, 2–275
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STOPTIME, 2–276
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STARTTIME, 2–277
TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STOPTIME, 2–277
TRIGGER:MAIN:VIDEO:HDTV, 2–278
TRIGGER:MAIN:VIDEO:HOLDOFF?, 2–279
TRIGGER:MAIN:VIDEO:HOLDOFF:VALUE, 2–279, 2–280
TRIGGER:MAIN:VIDEO:LINE, 2–280
TRIGGER:MAIN:VIDEO:LINES, 2–281
TRIGGER:MAIN:VIDEO:NTSC, 2–281
TRIGGER:MAIN:VIDEO:PAL, 2–282
TRIGGER:MAIN:VIDEO:SCAN, 2–283
TRIGGER:MAIN:VIDEO:SCANPERIOD, 2–283
TRIGGER:MAIN:VIDEO:SOURCE, 2–284
TRIGGER:MAIN:VIDEO:STANDARD, 2–284
TRIGGER:MAIN:VIDEO:SYNC, 2–285
TRIGGER:MAIN:VIDEO:SYSTEM, 2–286
TRIGGER:MAIN:VIDEO:TIME, 2–287
TRIGGER:STATE?, 2–287
*TST? query, 2–289

U
UNL, C–2
Unlisten, C–2
UNLOCK, 2–289
UNT, C–2
Untalk, C–2

V
VERBOSE, 2–290
Vertical
MATH<x>?, 2–168
MATH<x>:DEFINE, 2–168
MATH<x>:NUMAVg, 2–170
MATH<x>:PROCessing, 2–170
Index

Vertical bar cursors, 2–80
Vertical command group, 2–31
Vertical commands
  CH<x>?, 2–61
  CH<x>:BANDWIDTH, 2–61
  CH<x>:COUPLING, 2–62
  CH<x>:DESKEW, 2–63
  CH<x>:IMPEDANCE, 2–64
  CH<x>:OFFSET, 2–64
  CH<x>:POSITION, 2–66
  CH<x>:PROBE?, 2–66
  CH<x>:PROBECAL?, 2–67
  CH<x>:SCALE, 2–67
  CH<x>:VOLTS, 2–68
  SELECT?, 2–215
  SELECT:<wfm>, 2–216
  SELECT:CONTROL?, 2–217

W

*WAI, 2–290
Wait for operation complete, 2–290
Waveform, command mnemonic, 2–7
Waveform command group, 2–32
Waveform commands
  CURVE, 2–86
  DATA, 2–88
  DATA:DSTNATION, 2–88
  DATA:ENCODG, 2–89
  DATA:SOURCE, 2–91
  DATA:START, 2–92
  DATA:STOP, 2–93
  DATA:TARGET, 2–93
  DATA:WIDTH, 2–94
  WAVFRM?, 2–291
  WFMPRE?, 2–291
  WFMPRE:<wfm>?, 2–299
  WFMPRE:<wfm>:NR_PT, 2–299
  WFMPRE:<wfm>:PT_FMT, 2–300
  WFMPRE:<wfm>:PT_OFF, 2–301
  WFMPRE:<wfm>:WFID, 2–302
  WFMPRE:<wfm>:XINCR, 2–302
  WFMPRE:<wfm>:XUNIT, 2–303
  WFMPRE:<wfm>:XZERO, 2–303
  WFMPRE:<wfm>:YMULT, 2–304
  WFMPRE:<wfm>:YOFF, 2–304
  WFMPRE:<wfm>:YUNIT, 2–305
  WFMPRE:<wfm>:YZERO, 2–305
  WFMPRE:BTR BIT_NR, 2–292
  WFMPRE:BN_FMT, 2–293
  WFMPRE:BYT_NR, 2–293
  WFMPRE:BYT OR, 2–294
  WFMPRE:CRVCHK, 2–298
  WFMPRE:ENCODG, 2–295
  WFMPRE:NR_PT, 2–298
  WFMPRE:PT_FMT, 2–295
  WFMPRE:PT_OFF, 2–296
  WFMPRE:WFID, 2–298
  WFMPRE:XINCR, 2–297
  WFMPRE:XMULT, 2–298
  WFMPRE:XOFF, 2–298
  WFMPRE:XUNIT, 2–298
  WFMPRE:YMULT, 2–297
  WFMPRE:YOFF, 2–297
  WFMPRE:YUNIT, 2–298
  WFMPRE:YZERO, 2–298
  WAVFRM?, 2–291
  <wfm>, command mnemonic, 2–7
  WFMPRE?, 2–291
  WFMPRE:<wfm>?, 2–299
  WFMPRE:<wfm>:NR_PT, 2–299
  WFMPRE:<wfm>:PT_FMT, 2–300
  WFMPRE:<wfm>:PT_OFF, 2–301
  WFMPRE:<wfm>:WFID, 2–302
  WFMPRE:<wfm>:XINCR, 2–302
  WFMPRE:<wfm>:XUNIT, 2–303
  WFMPRE:<wfm>:XZERO, 2–303
  WFMPRE:<wfm>:YMULT, 2–304
  WFMPRE:<wfm>:YOFF, 2–304
  WFMPRE:<wfm>:YUNIT, 2–305
  WFMPRE:<wfm>:YZERO, 2–305
  WFMPRE:BTR BIT_NR, 2–292
  WFMPRE:BN_FMT, 2–293
  WFMPRE:BYT_NR, 2–293
  WFMPRE:BYT OR, 2–294
  WFMPRE:CRVCHK, 2–298
  WFMPRE:ENCODG, 2–295
  WFMPRE:NR_PT, 2–298
  WFMPRE:PT_FMT, 2–295
  WFMPRE:PT_OFF, 2–296
  WFMPRE:WFID, 2–298
  WFMPRE:XINCR, 2–297
  WFMPRE:XMULT, 2–298
  WFMPRE:XOFF, 2–298
  WFMPRE:XUNIT, 2–298
  WFMPRE:YMULT, 2–297
WFMPRE:YOFF, 2–297
WFMPRE:YUNIT, 2–298
WFMPRE:YZERO, 2–298
WFMPRE:ZMULT, 2–298
WFMPRE:ZOFF, 2–298
WFMPRE:ZUNIT, 2–298
WFMPRE:ZZERO, 2–298
Width trigger, 2–250, 2–265, 2–266, 2–267

Z

ZOOM, 2–306
Zoom command group, 2–37
Zoom commands
  ZOOM, 2–306
  ZOOM:DUAL, 2–306
  ZOOM:DUAL:OFFSET, 2–307
  ZOOM:GRATICULE, 2–308
  ZOOM:HORIZONTAL:LOCK, 2–308
  ZOOM:HORIZONTAL:POSITION, 2–309
  ZOOM:HORIZONTAL:SCALE, 2–310
  ZOOM:STATE, 2–310
  ZOOM:VERTICAL:POSITION, 2–311
  ZOOM:VERTICAL:SCALE, 2–312
  ZOOM GRATICULE, 2–308
  ZOOM:DUAL, 2–306
  ZOOM:DUAL:OFFSET, 2–307
  ZOOM:HORIZONTAL:LOCK, 2–308
  ZOOM:HORIZONTAL:POSITION, 2–309
  ZOOM:HORIZONTAL:SCALE, 2–310
  ZOOM:STATE, 2–310
  ZOOM:VERTICAL:POSITION, 2–311
  ZOOM:VERTICAL:SCALE, 2–312